SH 1 ÖS 100,–/Sfr. 14,– DM 14,–

64'er

1. PROGRAMM SONDERHEFT

Top-Listings zum Abtippen

TIPS & Tricks

30 Programme gleichzeitig im Speicher

Super-Disksorter

80 Zeichen auf dem Bildschirm

Befehlserweiterungen für Betriebssystem und Floppy

DATA-Maker und Dezimal-Disassembler

Viele tolle Programmierhilfen

Prüfsummen für jede Basic-Zeile

Alle Programme auch auf Diskette erhältlich!

Eine Markt & Technik Publikation

Warum ein Sonderheft?

Mit diesem Sonderheft für Tips & Tricks-Listings wollen wir dem Anwender des Commodore 64 und des VC 20 Programme in die Hand (im wahrsten Sinne des Wortes) geben, die ihm in der täglichen Arbeit mit dem Computer sicherlich bald unentbehrlich sein werden. Beim Abtippen kann praktisch kein Fehler mehr passieren, denn jede Zeile wird nach der Eingabe automatisch geprüft.

Die Frage, die sich so mancher Leser stellen wird, nämlich wozu dieses Sonderheft für Tips & Tricks-Listings dienen soll, läßt sich relativ einfach beantworten. All diese Programme wären sicherlich irgendwann einmal im 64'er Magazin für Commodore-Fans erschienen. Doch »leider« erscheint das 64'er nur jeden Monat einmal, und der Platz für Listings ist beschränkt. Es wäre also eine nicht unerhebliche Zeit verstrichen, bis alle Programme, die hier in kompakter und zu einem Thema gehörender Form zusammengefaßt sind, für Sie zum Abtippen und Anwenden bereitgestanden hätten. Also eine Energieleistung vollbracht und ein Sonderheft gemacht.

Die 64'er-Redaktion wäre aber nicht selbige, wenn dabei nicht irgendeine Neuerung oder Verbesserung herausspringen würde. So sind alle Listings in diesem Sonderheft mit einer zeilenweisen Prüfsumme versehen. Tippen Sie also zuallererst den Checksummer ein, und speichern Sie ihn ab. Nach dem Wiedereinladen und Initialisieren können Sie das von Ihnen gewünschte Listing eintippen, und zwar ohne Fehler. Sollten Sie in einer Zeile ein verkehrtes Zeichen eingegeben haben, so macht Sie der Checksummer sofort darauf aufmerksam. Die Fehlersuche durch das gesamte Programm können Sie also in Zukunft vergessen.

Aber auch die oft schwierig zu entziffernden Grafik- und Steuerzeichen wurden durch lesbare alphanumerische ersetzt. Also keine langwierige Suche mehr auf der Frontseite der Tastenkappen und Entscheidungsqualen, ob nun dicker oder halbdicker vertikaler Strich, kein nervenzermürbendes Nachschlagen in Tabellen, welches nun das Steuerzeichen für die Funktionstaste F7 ist — alles wird in Klartext ausgeben. Die genauen Spezifikationen zur Umsetzung der Grafik- und Steuerzeichen lesen Sie bitte in dem entsprechenden Artikel nach. Den Checksummer gibt es sowohl in einer Version für den C 64 als auch für den VC 20. Sicherlich, diese Methode des sicheren Eintippens ist nicht unsere Erfindung. Amerikanische Zeitschriften praktizieren sie seit einiger Zeit, doch dieses Programm ist eine Eigenentwicklung, und wir fordern Sie auf, Vergleiche anzustellen, es ist der besten eines.

Kommen weitere Sonderhefte?

Es wird nicht bei dieser einen Aktion bleiben, soviel können wir Ihnen versprechen. In der 64'er-Redaktion gehen täglich dutzende von Listings ein, für deren »Verwertung« in der Stammzeitschrift einfach nicht genug Platz vorhanden ist. Die genauen Erscheinungstermine liegen noch nicht fest, doch Sie können gespannt sein, was für Sonderhefte demnächst — in noch unregelmäßigen Abständen — erscheinen werden.

In unserem ersten Sonderheft sind 28 durchweg nützliche Programme enthalten. Es würde einige Zeit in Anspruch nehmen, alle selber einzutippen. Deshalb, wie auch in der 64'er Stammzeitschrift üblich, der Diskettenservice.

Disketten-Service

Programme, die sich auf die Floppy beziehen, also Dateiorganisation, Disksorter, ESF, Track 18, Diskettenmeister und Fileprotect, sind auf der Diskette CB 023 (Floppy-Utilities) für 29,90 Mark erhältlich.

Alle anderen auf der Diskette CB 024 (Hilfsprogramme) ebenfalls für 29,90 Mark.

Alle Listings wurden von den lauffähigen Versionen der Programme erstellt. Für das fehlerfreie Eintippen haben wir gesorgt. Dennoch kann es vorkommen, daß in einem Programm nach intensivem Ausnützen aller Funktionen und Möglichkeiten in dem einem oder anderen Teil ein logischer Denkfehler auftaucht. Für Hinweise die zur Ergreifung der Täter (Fehler) führen, ist zwar keine Belohnung ausgesetzt, aber sie werden im nächsten Sonderheft oder im 64'er veröffentlicht. Ebenso dankbar ist die Redaktion für Verbesserungs- und Ergänzungsvorschläge zu den einzelnen Listings wie zur gesamten Aufmachung des Sonderheftes.

Sollten Sie zu bestimmten Bereichen bessere Programme als die hier veröffentlichten geschrieben haben, so scheuen Sie sich nicht uns dies wissen zu lassen. Das nächste Sonderheft kommt bestimmt. Es gelten dieselben Einsendekriterien wie für das 64'er-Stammheft. (aa)

Inhalt

Checksummer — keine Fehler mehr beim Abtippen von Listings

Das Programm Checksummer 64 ist für all die Leute gedacht, die sich manchmal vor Verzweifelung »die Haare raufen« könnten. Da sitzt man mehrere Stunden, um ein gutes Programm aus dem 64'er Magazin abzutippen, und dann: ein Fehler in der DATA-Zeile oder ein falscher Buchstabe... und schon geht die Fehlersuche los. Hier soll der Checksummer 64 weiterhelfen.

Der Checksummer 64 ist ein kleines Maschinenprogramm, das, wenn es aktiviert ist, Sie sofort davon unterrichtet, ob Sie die jeweilige Programmzeile korrekt eingegeben haben.

1. Tippen Sie den Basic-Lader sorgfältig ein. Es gibt zwei Versionen: eine für den Commodore 64 und eine für den VC 20.

2. Bevor Sie »RUN« eingeben, speichern Sie den Basic-Lader bitte erst ab, denn wenn Sie zum Beispiel einen Fehler bei den eingetippten POKE-Anweisungen gemacht haben, ist es möglich, daß der Rechner aussteigt. Heben Sie sich den abgespeicherten Checksummer 64 auf — Sie werden ihn immer wieder brauchen, wenn Sie ein Basic-Programm aus dem 64'er Sonderheft oder in Zukunft aus dem 64'er Magazin eintippen wollen.

3. Der Checksummer 64 überprüft sich selbst. Wenn Sie einen Fehler in den DATAs gemacht haben, listen Sie die fehlerhafte Zeile einfach, korrigieren sie und starten dann das Programm neu.

4. Nach Initialisierung des Maschinenprogramms ist der Checksummer 64 aktiviert. Er steht innerhalb des Betriebssystems und verbraucht kein einziges Byte Speicherplatz. Es sei hier für Interessierte gesagt, daß selbst alle Sprungvektoren unverändert bleiben, das Programm also mit einer Vielzahl von anderen Programmier-Spracherweiterungen wie etwa Exbasic Level II problemlos zusammenarbeitet. Achten Sie aber darauf, daß bestimmte Spracherweiterungen das hinter dem ROM liegende RAM für Hires-Grafiken benutzen. Wird zum Beispiel eine Hires-Grafik von Simons Basic aus angesprochen, so wird der Checksummer 64 zerstört.

5. Wenn Sie den Checksummer 64 zwischenzeitlich nicht benutzen, können Sie ihn jederzeit mit »POKE 1, 55« desaktivieren. Auch durch Drücken der Run-Stop- und der Restore-Taste wird der Checksummer 64 desaktiviert. Wollen Sie, daß der Checksummer 64 auch noch nach Drücken dieser Tastenkombination erhalten bleibt, so geben Sie bei aktiviertem Checksummer 64 »POKE 64982, 53« ein. Der Checksummer 64 ist dann nur durch »POKE 1, 55« abschaltbar.

Wollen Sie den Checksummer 64 wieder einschalten, so geben Sie bitte »POKE 1, 53« ein.

Das Maschinenprogramm bleibt solange erhalten, bis der Computer ausgeschaltet , oder wenn von anderen Programmen auf das hinter dem ROM liegende RAM zugegriffen wird.

6. Eine Checksumme wird nur dann ausgegeben, wenn der Commodore 64 (VC 20) eindeutig erkennt, daß Sie eine Zeile bestehend aus der Zeilennummer und zumindestens einem alphanumerischen Zeichen eigegeben haben. Ansonsten reagiert der Commodore 64 normal.

Hinweis: Wenn Sie bei aktiviertem Checksummer 64 ein Programm mit »LOAD« in den Speicher holen, wird auch eine Checksumme ausgegeben. Dies liegt jedoch an rechnerinternen Routinen und hat keine weitere Bedeutung, stellt insbesondere keine Gefahr für das geladene Programm dar, da alle Pointer richtig gesetzt werden.

Nach Eingabe von RUN wird zunächst einmal das ROM in das RAM des Commodore 64 verschoben, wonach der Basic-Interpreter modifiziert wird. Dadurch hat man den Vorteil, trotz einer zusatzlichen Routine das gesamte RAM des Rechners zur Verfügung zu haben. Nach ordnungsgemässem Ablauf des Programms können Sie sofort mit Eingaben beginnen. Für Maschinensprache-Spezialisten weise ich darauf hin, daß ich ausnutze, daß die Einschaltmeldungen des Rechners nur nach einem Reset generiert wird. Der Textbereich, in dem die Meldung steht, wird von dem erzeugten Maschinenprogramm überschrieben.

Alle veröffentlichten Listings sind mit einer Checksumme versehen, die am Ende jeder Programmzeile steht. Diese Checksumme steht zwischen < und >. Sie wird beim Eintippen des Programms nicht mit eingegeben. Die Zahl zwischen den beiden Zeichen stellt lediglich eine Information für Sie dar. Wenn Sie diese Checksumme dennoch mit eintippen, werden Sie schnell bemerken, daß Sie etwas falsch gemacht haben. Bei aktiviertem Checksummer 64 wird nämlich nach Eingabe einer Basic-Zeile, die mit Return beendet wird, in die linke obere Bildschirmecke die Checksumme eingeblendet, die mit der Summe aus dem veröffentlichten Listing übereinstimmen muß. Ist das nicht der Fall, haben Sie die Zeile anders eingegeben, als sie im Listing dargestellt ist. Vergessen Sie also bitte nicht, daß die am Ende einer Zeile in < und > stehende Prüfsumme nicht mit eingegeben werden darf.

Der Checksummer 64 ist so ausgelegt, daß er abhängig von der Zeilennummer und dem Text der Zeile eine Checksumme ausgibt. Beim Bilden dieser Checksumme werden Spaces (Leertaste) überlesen, was für Sie bedeutet, das es egal ist, wieviel Leerzeichen Sie zwischen den Worten lassen, da Sie für den Programmablauf ohnehin keine Bedeutung haben. Aber manchmal ist das richtige Setzen von Leerzeichen doch wichtig, besonders innerhalb von Strings (Zeichenketten), die gedruckt werden sollen. Seien Sie deshalb besonders genau bei Leerzeichen, die innerhalb von Anführungszeichen stehen, denn meistens ermöglichen nur die richtig gesetzten Spaces eine sinnvolle Textausgabe auf dem Bildschirm.

Beachten Sie auch, daß es durchaus erlaubt ist, Abkürzungen für die Commodore-Befehlswörter zu verwenden. So führt die Eingabe von » ? « als Kurzschreibweise für »PRINT« nicht etwa zu einem Checksummen-Fehler, sondern wird korrekt verarbeitet und dementsprechend die Checksumme generiert. Nachdem Sie ein Listing eingegeben haben, sollten Sie es aus Sicherheitsgründen vor dem Starten abspeichern. Sie brau-

chen hierfür jedoch nicht den Checksummer 64 zu desaktivieren.

Hinweise zum Lesen von Listings

Die Listings haben sich ein wenig im Ausdruckformat verändert, um Ihnen das Eingeben von Programmen wesentlich zu erleichtern.

— Cursorsteuerzeichen und andere Steuerzeichen, die schwer zu lesen sind, werden von nun an in Klartext in speziellen Klammern gesetzt.

Tritt mehrmals hintereinander dasselbe Steuerzeichen auf, so wird diese Steuerzeichen-Sequenz zusammengefasst, indem zuerst das Steuerzeichen und dann die Anzahl der Wiederholungen dieses Steuerzeichens in Klartext ausgegeben wird.

— alle Commodore-Grafikzeichen, die über Shift zu erreichen sind, werden nicht mehr als Grafikzeichen, sondern als Klartextzeichen dargestellt. Dabei wird aus dem Zeichen, das Sie auf dem Bildschirm sehen, wenn Sie die Tastenkombination Shift und »A« ansprechen, wieder ein »A«. Um dieses »A« vom normalen »A« unterscheiden zu können, ist es etwas kleiner als das gewöhnliche »A« und ist außerdem mit einem Unterstreichungszeichen versehen. Diese Vereinbarung gilt auch für sämtliche andere Commodore-Grafikzeichen, die über Shift zu erreichen sind.

— entsprechendes gilt für sämtliche Commodore-Grafikzeichen, die über die Commodore-Taste zu erreichen sind. Hier wird jedoch das jeweilige Klartextzeichen nicht unterstrichen, sondern überstrichen.

Erläuterungen zu den Cursorsteuerzeichen

Cursorsteuerzeichen werden, wie schon oben erwähnt, umdefiniert. Sie sehen hier eine Liste der möglichen Ausdrücke, die für ein Cursorsteuerzeichen im Listing auftauchen können. Gleichzeitig ersehen Sie aus der Tabelle, welche Taste beziehungsweise Tastenkombination zu drücken ist, damit dieses Steuerzeichen richtig in Ihr Programm übernommen wird. Beachten Sie, daß Sie die Steuercodes nur dann als reverses Zeichen sehen können, wenn der Rechner im »Quote-Modus« arbeitet, das heißt er sich im Gänsefüßchenmodus befindet.

Wenn Sie lesen ! drücken Sie

CTRL steht für Control-Taste, so bedeutet[CTRL-A], daß Sie die Control-Taste und die Taste »A« drücken müssen. Im folgenden steht:

Wenn Sie lesen	! drücken Sie
[down]	! Taste neben rechtem Shift, Cursor unten
[up]	! Shift-Taste & Taste neben rechtem Shift, Cursor hoch
[clear]	! Shift-Taste & 2. Taste ganz rechts oben
[inst]	! Shift-Taste & Taste ganz rechts oben
[home]	! 2. Taste von ganz rechts oben
[del]	! Taste ganz rechts oben
[right]	! Taste ganz rechts unten
[left]	! Shift-Taste & Taste unten rechts
[space]	! Leertaste
[f1]	! grauer Tastenblock rechts
[f3]	! grauer Tastenblock rechts
[f5]	! grauer Tastenblock rechts
[f7]	! grauer Tastenblock rechts
[f2]	! grauer Tastenblock rechts & Shift
[f4]	! grauer Tastenblock rechts & Shift
[f6]	! grauer Tastenblock rechts & Shift
[f8]	! grauer Tastenblock rechts & Shift
[return]	! Shift-Taste & Return
[black]	! Control-Taste & 1
[white]	! Control-Taste & 2
[red]	! Control-Taste & 3
[cyan]	! Control-Taste & 4
[purple]	! Control-Taste & 5
[green]	! Control-Taste & 6
[blue]	! Control-Taste & 7
[yellow]	! Control-Taste & 8
[rvson]	! Control-Taste & 9
[rvoff]	! Control-Taste & 0
[orange]	! Commodore-Taste & 1
[brown]	! Commodore-Taste & 2
[lig.red]	! Commodore-Taste & 3
[grey 1]	! Commodore-Taste & 4
[grey 2]	! Commodore-Taste & 5
[lig.green]	! Commodore-Taste & 6
[lig.blue]	! Commodore-Taste & 7
[grey 3]	! Commodore-Taste & 8

Wenn Sie sich erst einmal an die in Klartext geschriebenen Steuerzeichen gewöhnt haben, werden Sie den Vorteil dieser Schreibweise erkennen. Der zu dem jeweiligen Steuerzeichen gehörende Klartext ist so verfaßt, daß Sie leicht die Taste beziehungsweise die Tastenkombination finden, die Sie drücken müssen.

Checksummer VC 20

Der Checksummer VC 20 ist im Prinzip genauso aufgebaut wie der Checksummer 64. Da beim VC 20 jedoch nicht die Möglichkeit besteht, das ROM softwaremässig zu modifizieren, mußte ein anderer Weg als beim Commodore 64 gewählt werden, um die Checksumme zu generieren.

In ihrer Funktionsweise unterscheiden sich der Checksummer VC 20 und der Checksummer 64 nicht. Es gelten folgende Sonderregelungen bei der Benutzung des Checksummer VC 20:

— da der Basic-Bereich nicht belegt werden soll, ist das Programm im Kassettenpuffer abgelegt.

— angeschaltet wird der Checksummer VC 20 mit »SYS 955«

— Abschaltung des Checksummer VC 20 wird mit »SYS 58459« vollzogen

ACHTUNG: Nehmen Sie keine Kassetten-Operationen vor, wenn der Checksummer VC 20 eingeschaltet ist. Da das Betriebssystem den Kassettenpuffer mit Daten belegt, kann der Checksummer VC 20 überschrieben werden, was zur Folge hat, daß sich der Rechner bei aktiviertem Checksummer VC 20 »aufhängt«. Wollen Sie deshalb ein Programm auf (von) Kassette abspeichern (laden), so müssen Sie erst den Checksummer VC 20 abschalten (SYS 58459).

Daraufhin kann der Kassettenpuffer mit Daten überschrieben werden, ohne daß der Rechner »aussteigt«.

Als Sicherung wird bei der Initialisierung geprüft, ob das zuletzt angesprochene Peripherie-Gerät der Kassettenrecorder war. Ist das der Fall, so werden die Betriebssystemroutinen LOAD und SAVE für die Benutzung gesperrt. Der Rechner meldet bei Aufruf einer dieser beiden Routinen READY, ohne weitere Aktionen durchzuführen. Diese Sicherung kann man nach der Tipparbeit aufheben, wenn man den Checksummer VC 20 mit SYS 58459 abschaltet. Dadurch wird der Kassettenpuffer für andere Daten freigemacht. Weiterhin wird dann durch gleichzeitiges Drücken der Tasten »Run-Stop & Restore« erreicht, daß die Betriebssystemroutinen LOAD und SAVE wieder eingerichtet werden.

— Bei Benutzung einer Diskettenstation brauchen Sie nicht darauf zu achten, daß bei LOAD beziehungsweise SAVE der Checksummer VC 20 überschrieben wird, da der Kassettenpuffer für die Diskettenstation normalerweise nicht genutzt wird. Deshalb können Sie die beiden Routinen weiterhin normal nutzen, sofern der Rechner bei der Initialisierung des Checksummer VC 20 feststellt, daß das zuletzt angesprochene Peripherie-Gerät nicht der Kassettenrecorder war.

— bedingt durch den anderen Aufbau des Checksummer VC 20 wird anders als beim Checksummer 64 nach der LOAD-Routine keine Checksumme ausgegeben.

— wird eine Zeile gelöscht, also eine Zahl zwischen 0 und 63999 eingegeben, und danach Return gedrückt, so wird eine Checksumme ausgegeben, die aber keine Bedeutung hat.

Viel Spaß beim Eintippen von Programmen mit dem neuen Checksummer!

(F. Lonczewski / gk)

```
10 REM *******************************      <175>
20 REM *                             *      <247>
30 REM *       CHECKSUMMER 64        *      <162>
40 REM *                             *      <011>
50 REM *           64'ER             *      <061>
60 REM *                             *      <031>
70 REM *        COMMODORE 64         *      <056>
80 REM *                             *      <051>
90 REM *******************************      <255>
100 PRINT"[CLEAR,SPACE13,RVSON]CHECKSUMMER
   [SPACE]64[RVOFF]"                        <025>
110 PRINT                                   <007>
120 PRINT"[DOWN2,SPACE4]ICH[SPACE]ARBEITE!
   [SPACE]BITTE[SPACE]ETWAS[SPACE]GEDULD." <071>
130 FOR I=40960 TO 49151:POKE I,PEEK(I):NEXT
                   <007>
140 FOR I=57344 TO 65535:POKE I,PEEK(I):NEXT
                   <025>
150 POKE 1,53:POKE 42289,96:POKE 42290,228 <001>
160 FOR I=58464 TO 58554:GOSUB 220:POKE I,A
                   <184>
170 PS=PS+A+1:NEXT I                        <109>
180 IF PS<>11187 THEN PRINT"PRUEFSUMMENFEHLER
   [SPACE]!":END                            <180>
190 PRINT"[DOWN4,SPACE9]CHECKSUMMER[SPACE]
   AKTIVIERT."                              <247>
200 PRINT"[DOWN2]AUSSCHALTEN[SPACE]
   :[SPACE]POKE1,55"                        <050>
210 PRINT"[DOWN]ANSCHALTEN[SPACE2]
   :[SPACE]POKE1,53":NEW                    <171>
220 READ A$:IF LEN(A$)<>2 THEN PRINT"TIPPFEHLER
   [SPACE]IN[SPACE]ZEILE"PEEK(63)+PEEK(64)*256
   :END                                     <201>
230 A1=ASC(A$):A2=ASC(RIGHT$(A$,1))         <216>
240 IF A1<48 OR A1>57 THEN IF A1<65 OR A1>70 TH
   EN 310                                   <130>
250 IF A2<48 OR A2>57 THEN IF A2<65 OR A2>70 TH
   EN 310                                   <144>
260 IF A1>64 THEN A1=A1-55:GOTO 280         <204>
270 IF A1<58 THEN A1=A1-48                  <128>
280 IF A2>64 THEN A2=A2-55:GOTO 300         <220>
290 IF A2<58 THEN A2=A2-48                  <151>
300 A=A1*16+A2:RETURN                       <138>
310 PRINT"UNGUELTIGER[SPACE]HEXCODE[SPACE]IN
   [SPACE]ZEILE"PEEK(63)+PEEK(64)*256:END  <021>
320 DATA A0,02,A9,00,85,02,B1,5F            <098>
330 DATA F0,0F,C9,20,D0,03,C8,D0            <146>
340 DATA F5,18,65,02,85,02,4C,6E            <126>
350 DATA E4,C0,04,30,F1,C6,D6,A5            <169>
360 DATA D6,48,A2,03,A9,20,9D,01            <150>
370 DATA 04,BD,B7,E4,20,D2,FF,CA            <238>
380 DATA 10,F2,A6,02,A9,00,20,CD            <169>
390 DATA BD,A9,3E,20,D2,FF,68,85            <245>
400 DATA D6,20,6C,E5,A9,8D,20,D2            <229>
410 DATA FF,4C,80,A4,5C,48,20,C9            <244>
420 DATA FF,AA,68,90,01,8A,60,09            <234>
430 DATA 3C,12,13                           <199>
```

```
10 REM****************                      <057>
20 REM*               *                     <247>
30 REM* CHECKSUMMER   *                     <056>
40 REM* VERSION VC20 *                      <044>
50 REM*               *                     <021>
60 REM****************                      <107>
70 PRINT"[CLEAR,SPACE2,RVSON]CHECKSUMMER[SPACE2]
   VC-20[RVOFF]"                            <185>
80 PRINT                                    <233>
90 PRINT"[DOWN]EINEN[SPACE]MOMENT,[SPACE]
   BITTE..."                                <181>
100 FOR I=827 TO 993:GOSUB 180:POKE I,A     <177>
110 PS=PS+A+1:NEXT I                        <049>
120 IF PS<>20612 THEN PRINT"[DOWN]
   PRUEFSUMMENFEHLER[SPACE]!":END           <130>
130 SYS 955:PRINT"CHECKSUMMER[SPACE]AKTIVIERT."
                   <242>
140 PRINT"AN[SPACE]:SYS955"                 <212>
150 PRINT"[DOWN]AUS:SYS58459,[SPACE]BEI[SPACE]
   CAS-[SPACE4]SETTE[SPACE]ZUSAETZLICH[SPACE5]
   RUN/STOP[SPACE]&[SPACE]RESTORE"          <068>
160 PRINT"[DOWN]BEI[SPACE]AKTIVIERTEM[SPACE]
   CHECK-SUMMER[SPACE]KEIN";                <105>
170 PRINT"[SPACE]CASSETTEN-BETRIEB[SPACE](LOAD,
   [SPACE]SAVE)[SPACE2]ERLAUBT!":NEW        <051>
180 READ A$:IF LEN(A$)<>2 THEN PRINT"TIPPFEHLER
   [SPACE]IN[SPACE]ZEILE"PEEK(63)+PEEK(64)*256
   :END                                     <161>
```

```
190 A1=ASC(A$):A2=ASC(RIGHT$(A$,1))                <176>
200 IF A1<48 OR A1>57 THEN IF A1<65 OR A1>70 TH
   EN 270                                          <095>
210 IF A2<48 OR A2>57 THEN IF A2<65 OR A2>70 TH
   EN 270                                          <109>
220 IF A1>64 THEN A1=A1-55:GOTO 240                <159>
230 IF A1<58 THEN A1=A1-48                         <087>
240 IF A2>64 THEN A2=A2-55:GOTO 260                <184>
250 IF A2<58 THEN A2=A2-48                         <110>
260 A=A1*16+A2:RETURN                              <098>
270 PRINT"UNGUELTIGER[SPACE]HEXCODE[SPACE]IN
   [SPACE]ZEILE"PEEK(63)+PEEK(64)*256:END          <237>
280 DATA 20,5F,03,86,7A,84,7B,20                   <061>
290 DATA 73,00,AA,F0,F3,A2,FF,86                   <130>
300 DATA 3A,90,0A,A2,00,86,FF,20                   <097>
310 DATA 79,C5,4C,E1,C7,A2,01,86                   <127>
320 DATA FF,4C,9C,C4,A6,FF,E0,01                   <199>
330 DATA F0,03,4C,60,C5,A0,02,A9                   <125>
340 DATA 00,85,FE,B1,5F,F0,0F,C9                   <186>
350 DATA 20,D0,03,C8,D0,F5,18,65                   <141>
360 DATA FE,85,FE,4C,76,03,C0,04                   <193>
370 DATA 30,F1,C6,D6,A5,D6,48,A2                   <198>
380 DATA 03,A9,20,9D,01,04,BD,B7                   <180>
390 DATA 03,20,D2,FF,CA,10,F2,A6                   <217>
400 DATA FE,A9,00,20,CD,DD,A9,3E                   <016>
410 DATA 20,D2,FF,68,85,D6,20,87                   <220>
420 DATA E5,A9,8D,20,D2,FF,A2,00                   <003>
430 DATA 86,FF,F0,AE,09,3C,12,13                   <002>
440 DATA A9,3B,8D,02,03,A9,03,8D                   <249>
450 DATA 03,03,A5,BA,C9,01,D0,10                   <235>
460 DATA A9,74,8D,30,03,8D,32,03                   <239>
470 DATA A9,C4,8D,31,03,8D,33,03                   <007>
480 DATA AD,88,02,8D,90,03,60                      <113>
```

M-P-S: Multi-Programm-System Mehr als 30 Programme gleichzeitig im Speicher

Wer hat nicht schon den Wunsch verspürt, mehr als ein Programm gleichzeitig im Speicher zu haben. M-P-S erlaubt sogar bis zu 32 verschiedene Basic-Programme. Dafür stehen 42 KByte(!) freier Basic-Speicher zur Verfügung. Jedes Programm kann unabhängig von einem anderen aufgerufen und gestartet werden. Zu jeder Zeit behalten Sie den Überblick über die im Computer stehenden Programme. Auch Directories lassen sich permanent im Speicher halten. Eine tolle Sache.

M-P-S bietet die Möglichkeit, mehr als nur ein Basic-Programm im Speicher zu halten. Vorgesehen ist eine Verwaltung von maximal 32 Basic-Programmen, wobei die Anzahl der tatsächlich speicherbaren Programme aber auch von deren Umfang abhängt. Jedes Programm erhält eine Kennummer, die zusammen mit den Programmnamen, Anfangs- und Endadresse in einem Inhaltsverzeichnis abgelegt wird. Durch die Eingabe der Kennummer kann dann jedes einzelne Basic-Programm aufgerufen werden. Für die Basic-Programme stehen etwa 42-KByte zur Verfügung. Dies wurde dadurch erreicht, daß Tabellen und Programmspeicher teilweise in den RAM-Bereich unter dem ROM gelegt wurde.

Um die Länge von M-P-S in Grenzen zu halten, wurde die Länge der Befehlswörter auf drei Buchstaben begrenzt.

Die Befehle:

1. OFF
2. STO (Store)
3. REC (Recall)
4. DSP (Display)
5. SPA (Space)
6. DEL (Delete)

1. OFF

OFF bewirkt ein Abschalten von M-P-S. Möchte man anschließend M-P-S wieder aktivieren, so gibt es zwei Möglichkeiten:

a) Die Eingabe von SYS 51200 bewirkt einen Kaltstart. Alle Pointer werden neu gesetzt. Das Inhaltsverzeichnis wird gelöscht.
b) Die Eingabe von SYS 51297 bewirkt einen Warmstart. Inhaltsverzeichnis und Pointer bleiben erhalten.

2. STO"Programmname",X

Mit STO wird ein Basic-Programm in einen bestimmten RAM-Bereich abgespeichert (verschoben). Der Programmname darf maximal 16 Zeichen lang sein. X ist eine Zahl zwischen 0 und 31. Sie dient als Trennmarke für die einzelnen Programme. Am besten fängt man bei 0 an und erhöht X mit jedem weiteren Programm um 1.
Beispiel:
STO"Test 1",0
STO"Test 2",1
STO"Test 3",2

3. RECN

Mit REC kann ein abgespeichertes Programm aufgerufen werden. N ist hierbei die Kennummer (siehe 4. DSP). Anschließend kann man LIST oder RUN eingeben.
Beispiel:
REC0
REC1
REC2

4. DSP

DSP gibt das Inhaltsverzeichnis der im Speicher stehenden Programme auf dem Bildschirm aus. Dabei erscheinen Kennnummer, Programmname und belegter Speicherbereich.
Beispiel:

0.	»Test 1«	\$C7FF-	\$C600
↑	↑	↑	↑
Kennummer	Name	Anfangsadresse	Endadresse+1

5. SPA

SPA gibt die Anfangs(immer \$0801)- und Endadresse des gerade lauffähigen Basicgrogramms, sowie dessen Länge in Bytes an. Man kann so feststellen, ob das Programm noch verschoben werden kann.
Beispiel (nach NEW):
SPA
\$0801-\$0803 / 3 Bytes

6. DEL

DEL löscht das Inhaltsverzeichnis und setzt die Pointer neu. Solange man mit STO keine neuen Programme eingibt, kann man die alten noch mit REC aufrufen.

Hinweise:

Folgendes sollte bei der Benutzung von M-P-S beachtet werden:
1. Die Basic-Programme, die abgespeichert und später auch gestartet werden sollen, dürfen keine Maschinenprogramme enthalten. Ansonsten besteht die Möglichkeit, daß das Programm M-P-S selbst, Tabellen oder Basic-Programme überschrieben werden und so nicht mehr lesbar sind.
2. Bei REC dürfen nur tatsächlich vorhandene Kennummern eingegeben werden. Gibt man Nummern ein, zu denen kein Programm vorhanden ist, so führt dies meist zum Systemabsturz.

Adressenbelegung:

M-P-S belegt: \$C800-\$CC8F (1167 Bytes)
Befehl OFF:\$C8F1-\$C8FD
SPA:\$C930-\$C97E
STO:\$C97F-\$CA77
DSP:\$CA78-\$CB4D
REC:\$CB4E-\$CBDD
DEL:\$CC24-\$CC5C

Arbeitsspeicher und Pointer:\$CC90-\$CC99

Speicher für Programmnamen ab:\$F000
Tabelle der Anfangs- und Endadressen ab:\$F800

Speicher für Basic-Programme ab:\$C7FF (abnehmend)

(Kasem Mossavi / gk)

```
0 REM  ********************           <215>
1 REM  *        M-P-S      *           <046>
2 REM  *        -----      *           <198>
3 REM  *  K.MOSSAVI        *           <129>
4 REM  *  KIRCHWEG 24      *           <161>
5 REM  *  3008 GARBSEN 1   *           <230>
6 REM  *  TEL.05137/71767  *           <055>
7 REM  ********************           <222>
8 :                                    <066>
9 :                                    <067>
10 PRINT"[CLEAR]MOMENT[SPACE]BITTE..." <076>
20 FOR I=51200 TO 52367                <051>
30 READ A:POKE I,A:L=L+A               <220>
40 NEXT I                              <243>
50 IF L<>157782 THEN PRINT"FEHLER[SPACE]IN
   [SPACE]DATAS[SPACE]!":END           <196>
60 SYS 51200                           <210>
61 DATA 169,0,141,32,208,141,33,208,169,1,141,
   134,2,169,23,160,200,32,30          <250>
62 DATA 171,76,93,204,147,32,32,32,32,32,32,42,
   42,42,32,77,85,76,84,73,45          <081>
63 DATA 80,82,79,71,82,65,77,77,45,83,89,83,84,
   69,77,32,42,42,42,13,13,32          <118>
64 DATA 32,32,32,32,32,32,32,32,32,40,67,41,32,
   66,89,32,75,46,77,79,83,83          <083>
65 DATA 65,86,73,32,32,49,57,56,52,13,13,13,0,
   169,110,160,200,141,4,3,140         <053>
66 DATA 5,3,76,116,164,162,0,160,0,185,0,2,221,
   190,200,240,17,189,190,200          <046>
67 DATA 208,3,76,187,200,185,0,2,232,232,232,76,
   117,200,232,200,185,0,2,221         <145>
68 DATA 190,200,240,5,202,136,76,130,200,232,
   200,185,0,2,221,190,200,240         <235>
69 DATA 7,202,202,136,136,76,130,200,189,215,
   200,141,185,200,232,189,215         <017>
70 DATA 200,141,186,200,76,48,201,76,124,165,79,
   70,70,83,80,65,83,84,79,68          <163>
71 DATA 83,80,82,69,67,68,69,76,69,88,67,0,0,0,
   0,0,0,241,200,0,48,201,0,127        <175>
72 DATA 201,0,120,202,0,78,203,0,36,204,0,99,
   204,0,255,255,0,169,124,160         <250>
73 DATA 165,141,4,3,140,5,3,76,116,164,72,74,74,
   74,74,32,22,201,170,104,41          <117>
74 DATA 15,32,22,201,72,138,32,210,255,104,76,
   210,255,24,105,246,144,2,105        <106>
75 DATA 6,105,58,96,169,36,32,210,255,165,253,
   32,254,200,165,254,32,254,200       <188>
76 DATA 96,169,13,32,210,255,169,1,162,8,133,
   254,134,253,32,32,201,169,45        <090>
77 DATA 32,210,255,165,45,166,46,133,254,134,
   253,32,32,201,169,32,32,210         <026>
78 DATA 255,169,47,32,210,255,169,32,32,210,255,
   165,46,56,233,8,133,98,165          <162>
79 DATA 45,133,99,32,209,189,169,119,160,201,32,
   30,171,76,116,164,32,66,89          <166>
80 DATA 84,69,83,13,0,230,122,230,122,230,122,
   169,34,32,255,174,174,144,204       <178>
81 DATA 172,145,204,232,208,1,200,142,164,201,
   140,165,201,160,0,185,4,2,201       <145>
82 DATA 34,240,16,141,11,240,238,164,201,208,3,
   238,165,201,200,192,16,208          <070>
83 DATA 233,152,24,101,122,133,122,230,122,32,
   253,174,32,158,183,224,32,144       <165>
84 DATA 3,76,72,178,173,164,201,172,165,201,141,
```

```
   213,201,140,214,201,142,11                <117>
85 DATA 240,169,1,141,238,201,169,8,141,239,201,
   173,146,204,174,147,204,141               <192>
86 DATA 241,201,142,242,201,173,249,10,141,68,
   194,238,238,201,208,3,238,239             <195>
87 DATA 201,206,241,201,208,3,206,242,201,173,
   238,201,197,45,208,227,173                <036>
88 DATA 239,201,197,46,208,220,174,148,204,172,
   149,204,173,146,204,142,33                <107>
89 DATA 202,140,34,202,141,4,248,232,208,1,200,
   173,147,204,142,49,202,140                <066>
90 DATA 50,202,141,5,248,232,208,1,200,173,241,
   201,142,65,202,140,66,202                 <015>
91 DATA 141,6,248,232,208,1,200,173,242,201,142,
   81,202,140,82,202,141,7,248               <172>
92 DATA 232,208,1,200,142,148,204,140,149,204,
   173,241,201,174,242,201,141               <072>
93 DATA 146,204,142,147,204,173,164,201,174,165,
   201,141,144,204,142,145,204               <186>
94 DATA 76,116,164,169,13,32,210,255,169,1,162,
   240,141,161,202,142,162,202               <139>
95 DATA 162,248,142,208,202,162,0,138,141,207,
   202,72,169,0,32,205,189,32                <046>
96 DATA 222,203,120,169,53,133,1,173,1,240,162,
   55,134,1,88,201,255,208,3                 <035>
97 DATA 76,61,203,201,32,144,14,32,210,255,238,
   161,202,208,3,238,162,202                 <035>
98 DATA 76,155,202,169,34,32,210,255,32,238,203,
   120,169,53,133,1,173,0,248                <151>
99 DATA 133,254,174,207,202,172,208,202,232,208,
   1,200,142,228,202,140,229                 <086>
100 DATA 202,173,5,248,133,253,32,247,203,169,
   45,32,14,204,174,228,202,172              <108>
101 DATA 229,202,232,208,1,200,142,1,203,140,2,
   203,173,6,248,133,254,174,1               <076>
102 DATA 203,172,2,203,232,208,1,200,142,22,203,
   140,23,203,173,7,248,133,253              <171>
103 DATA 232,208,1,200,142,207,202,140,208,202,
   32,247,203,104,168,200,152                <026>
104 DATA 72,170,169,13,32,5,204,238,161,202,208,
   3,238,162,202,76,147,202,169              <213>
105 DATA 20,32,210,255,32,210,255,32,210,255,32,
   210,255,76,116,164,32,84,203              <188>
106 DATA 76,116,164,230,122,230,122,230,122,32,
   158,183,134,251,169,4,133,252             <202>
107 DATA 169,0,133,253,162,8,70,252,144,3,24,
   101,251,106,102,253,202,208,243           <191>
108 DATA 165,253,141,136,203,169,248,141,137,
   203,160,0,120,169,53,133,1,185            <162>
109 DATA 0,248,153,150,204,200,192,4,208,245,
   169,54,133,1,88,173,150,204,141           <208>
110 DATA 174,203,173,151,204,141,175,203,169,1,
   141,177,203,169,8,141,178,203             <220>
111 DATA 173,60,197,141,196,10,206,174,203,208,
   3,206,175,203,238,177,203,208             <226>
112 DATA 3,238,178,203,173,174,203,205,152,204,
   208,226,173,175,203,205,153               <122>
113 DATA 204,208,218,173,177,203,133,45,32,28,
   204,133,46,96,169,46,32,210               <081>
114 DATA 255,169,32,32,210,255,169,34,32,210,
   255,96,169,32,32,210,255,32,210           <221>
115 DATA 255,96,169,55,133,1,88,32,32,201,120,
   169,53,133,1,96,160,55,132,1              <126>
116 DATA 88,32,210,255,96,160,55,132,1,88,32,
   210,255,120,169,53,133,1,96,160           <227>
117 DATA 55,132,1,173,178,203,96,32,42,204,76,
   116,164,169,0,141,144,204,141             <172>
118 DATA 148,204,169,240,141,145,204,169,255,
   141,146,204,169,199,141,147,204           <246>
119 DATA 169,248,141,149,204,160,0,169,255,153,
   0,240,153,0,241,153,0,242,153             <214>
120 DATA 0,243,153,0,244,200,208,238,96,32,42,
   204,76,107,204,32,84,203,169              <121>
121 DATA 138,76,231,167,162,1,160,6,24,32,240,
   255,160,0,169,61,32,210,255               <071>
122 DATA 200,192,29,208,248,169,13,32,210,255,
   32,210,255,32,210,255,32,210              <114>
123 DATA 255,76,97,200                       <141>
```

Datei-organisation

Dieses Programm ist eine echte Hilfe für den C64-Benutzer, der sich einmal etwas näher mit der internen Dateiorganisation des 1541-Laufwerks beschäftigen möchte.

Die Idee zu diesem Programm kam mir, als ich nach einem »SCRATCH«-Befehl mit Bedauern feststellen mußte, ein in mühevoller Arbeit erstelltes Programm versehentlich mitgelöscht zu haben. Doch glücklicherweise hatte ich kurz zuvor erfahren, wie man ein gelöschtes File auf der 1541-Floppy regeneriert: Filetyp im Directoryeintrag im Direktzugriff wieder setzen und anschließend ein »VALIDATE« ausführen.

Es begann also die Suche nach besagtem Byte, sowie die intensive Erkundung der Dateiorganisation und des Direktzugriffs.

Nun, nach mehreren Monaten Arbeit liegt endlich das fertige Programm vor. Es druckt eine Vielzahl von Floppy-Informationen auf einem Epson-Drucker aus. Auf eine Bildschirmausgabe wurde aus Gründen der Übersichtlichkeit verzichtet.

Was bietet das Programm

Ganze Dateien aufzeigen:

Hierbei fragt das Programm nach dem gewünschten Filenamen. Bei korrekter Eingabe werden folgende Informationen ausgedruckt: Filename, Filetyp, Filelänge in Blocks, Spur, Sektor, laufende Blocknummer und natürlich der Inhalt sämtlicher von dieser Datei belegten Blöcke in ASCII-Codes und -Zeichen (je 8 Byte). Eine Seite wird mit genau 2 Blocks beschrieben, dann erfolgt automatisch ein Seitenvorschub. Bei ungerader Blockanzahl kann ein Seitenvorschub auch vom Benutzer ausgelöst werden. Bei Dateiende springt das Programm wieder ins Menü zurück.

Einzelne Blöcke ausgeben:

Dieser Modus unterscheidet sich vom vorigen dadurch, daß hier bestimmte Sektoren ausgewählt werden können. Und dies auf zwei verschiedene Arten:

1. Eingabe von Spur und Sektor:
   Geben Sie, durch Komma getrennt, Spur- und Sektornummer des auszudruckenden Blocks ein;
2. Eingabe von Filename und Blocknummer: Das Programm liest aus der Directory, wieviele Blocks das angegebene File belegt und zeigt dies am Bildschirm an. Nun können Sie wählen, welchen dieser Blöcke Sie sehen möchten.

Bytereihen ausdrucken:

Mit diesem Menüpunkt können Sie gezielt bestimmte Bytes einer Datei oder eines Sektors herauspicken. Die Form der Eingabe von Parametern ist identisch mit der des Teilprogrammes »Einzelne Blöcke ausgeben«. Zusätzlich müssen Sie das erste und letzte auszudruckende Byte eines Sektors festlegen (0-255). Das erste Byte kann auch zugleich das letzte sein, was die Ausgabe nur eines Wertes bewirkt.

Blockbelegung von Files:
Von einer Datei belegte Sektoren werden ihrer Reihenfolge entsprechend, im Format: laufende Blocknummer, Spur/Sektor, ausgegeben. Auch eine Routine zur Berechnung der tatsächlichen Filelänge in Bytes ist in dieses Teilprogramm integriert. Das zweite Byte des letzten Blockes gibt die Anzahl der von der Datei in diesem Block benötigten Bytes an. Daraus ergibt sich folgende Formel für die Filelänge: (Gesamtzahl belegter Blocks — 1) * 254 + 2. Byte letzter Block.

Einschränkung für relative Dateien: Spur und Sektor von Side-Sektor-Blöcken werden nicht angezeigt. An ihrer Stelle erscheint das Kürzel »SSB«, so daß man wenigstens weiß, wieviel SSB das DOS zur Verwaltung der Datei benötigt.

Directory:
Dieser Befehl bringt Ihnen eine erweiterte und vor allem übersichtlichere Directory Ihrer Disketten zu Papier. Selbstverständlich ohne das Programm zu zerstören. Sie enthält Informationen über:
— Diskettenname und ID
— Nummer des Eintrages
— Filename
— Filelänge in Blocks
— Filelänge in Bytes = (Filelänge in Blocks — 1) * 254 + 127)

Es wird davon ausgegangen, daß der letzte Block zur Hälfte belegt ist. Die Angabe weicht also um maximal 126 Bytes von der tatsächlichen Länge ab.
— 1. Datenblock des Files
— Filetyp
— Freie Einträge
— Freie Blöcke

Bei relativen Dateien finden Sie im Anschluß an den Filetyp zwei zusätzliche Angaben:
— Erster Side - Sektor - Block (Spur/Sektor)
— Recordlänge

BAM (Block Available Map)
Das Format des Blockbelegungsplanes ist wie folgt festgelegt: Horizontal werden die einzelen Sektoren aufgetragen, vertikal die Spuren 1—35. Ein Stern kennzeichnet einen belegten Sektor, das Minuszeichen einen freien. Mit »/« markierte Sektoren sind nicht vorhanden, da die Anzahl der Sektoren je Spur zum Diskettenmittelpunkt hin abnimmt. Dem Wert rechts neben jeder Spur können Sie entnehmen, wieviele Blöcke auf dieser noch nicht belegt sind. Die Summe der freien Sektoren finden Sie ganz unten. Sollte einmal die Gesamtzahl der freien Blöcke nicht mit der Angabe in der Directory übereinstimmen, so ist mit der Diskette ein »VALIDATE« durchzuführen, um den korrekten Zustand wieder herzustellen.

Benötigte Hardware

Erstellt und getestet wurde das Programm auf einem C 64 mit Floppy VC 1541 und Epson RX-80 (mit Data-Becker-Interface).

Eine Umsetzung auf andere Druckertypen dürfte keine großen Schwierigkeiten machen, da die meisten Steuerzeichen für Druckerfunktionen in den Zeilen 4220-4370 untergebracht sind. Weitere Steuerzeichen, welche vereinzelt im Programmtext vorkommen, sind durch die Form (c$="string":c3=Zahl:gosub4030) leicht zu erkennen. Durch diese Druckparameter werden vor allem Schriftart und Zeilenabstand gesteuert.

Ausnahme: Teilprogramm »Blockbelegung von Files«
Zeilen 1890—1900 : A(3) = 254
1930—1940
2020—2030 : A(3) = 255

In diesen Zeilen wird in den Grafikmodus des Druckers umgeschaltet und eine Punktmatrix (CHR$(254) oder CHR$(255) für die senkrechten Linien, CHR$(64) für die waagrechte Linie übermittelt.

Mit diesen Hinweisen sollte die Anpassung an andere Drucker keine Schwierigkeit mehr darstellen.

Zum Schluß möchte ich noch auf ein Buch hinweisen, das mir erst die nötigen Grundlagen zu diesem Programm vermittelte: »Das große Floppy-Buch« von Data Becker. Eine wirklich ausgiebige und informative Quelle für alle jene, die sich intensiver mit der Floppy-Programmierung befassen wollen.

Ich denke, daß Sie das Programm nun sinnvoll einsetzen können, zumal es auch Routinen enthält, die man in andere Programme übernehmen kann.

(Johann Auer / ev)

Tabelle 1. Programmbeschreibung

Zeile(n)	Bedeutung
10-100	Titelvorspann
170	Aufruf 'Initialisierung'
190	Aufruf 'Menü'
210	Sprungbefehl zu Teilprogrammen (Startadressen)
280-430	Initialisierung
460-770	Menü
710-750	Prüfen ob Drucker eingeschaltet
800-960	Ganzes File ausgeben (Teilprogramm)
990-1350	Einzelne Blöcke ausdr. (Teilprogramm)
1050-1080	Eingabewahl: 1) Spur/Sektor 2) Filename/Blocknr.
1380-1680	Bytereihen (Teilprogramm)
1440	Sprung zu Eingabewahl (Zeilen 1050-1080)
1710-2130	Blockbelegung von Files (Teilprogramm)
2050	Spur/Sektor des nächsten Datenblocks laden
2160-2870	Direktory aufzeigen (Teilprogramm)
2430-2650	8 mal 32 Zeichen eines Directoryeintrages lesen — Zerlegung dieses Strings
2900-3430	BAM (Teilprogramm)
3150-3220	4 Bytes für je eine Spur lesen und in Binärstring umwandeln
3390-3430	Anzahl Sektoren pro Spur berechnen
3460-3490	Ende (Teilprogramm)

3520-3760	File vorhanden (Directory durchsuchen)
3780-3790	Floppy initialisieren
3810-3850	Sektor in Pufferspeicher laden — Pufferzeiger setzen
3870-3910	2 Bytes aus Pufferspeicher holen
3930-3940	Floppykanäle schließen
3960-3990	Drucker initialisieren
3980	Drucker auf halbe Geschwindigkeit stellen (kann weggelassen werden)
4010-4030	Parameterisierte Schriftenübermittlung zum Drucker
4050-4200	Kopfzeilen für Menüpunkte 1—4 drucken
4220-4370	Verschiedene Druckparameter für Schriftart
4390-4630	Bytenummer, Bytes und ASCII-Zeichen für Teilprogramme 1—3 drucken
4490-4560	8 Zeichen aus Pufferspeicher holen und umwandeln
4650-4750	Basic-Token der Operatoren in Zeichen umwandeln und String aufaddieren
4770-4830	Filename eingeben
4840-4890	Hinweis, wenn File nicht vorhanden
4910-4950	Für Teilprogramm 1 prüfen, ob bereits 2 Sektoren pro Seite gedruckt sind
4970-5040	Eingabe Blocknummer für Teilprogramm 2 und 3, Laden dieses Blocks
5060-5090	Abfrage Seitenvorschub
5110-5160	Tastaturabfrage auf J/N
5180-5210	F1 - Funktionstastenabfrage
5230-5260	Aufforderung gewünschte Diskette einzulegen
5280-5300	F1 - Fertig - Abfrage
5320-5410	Diskettenname und ID lesen

Tabelle 2. Unterprogramme

Name	Bedeutung
NU$	String mit 8 Nullen und 8 Punkten
A	Teilprogrammnummer
FT$(I)	Filetypen
BN	aktuelle Blocknummer
AN	Anfangswert für Laufvariable in UP »Bytes ausdrucken«
EN	Endwert für Laufvariable in UP »Bytes ausdrucken«
BP	Block Pointer (Zeiger im Pufferspeicher)
SP	Spur
SE	Sektor
S1/S2	ASC-Codes aus UP »2 Bytes holen«
MSE	Maximale Anzahl Sektoren pro Spur
EB	Erstes zu druckendes Byte in Teilprogramm »Bytereihen«
LB	Letztes zu druckendes Byte in Teilprogramm »Bytereihen«
RE	Anzahl Blöcke pro Zeile bei Teilprogramm 4
AR	Anzahl der Reihen in Teilprogramm 4
FL	Filelänge in Blocks
DN$	enthält den Diskettennamen
ID$	ID der Diskette
FB	freie BLöcke
ET$	ein kompletter Directoryeintrag
EN$	Eintragsnummer
FT$	Filetyp (String)
FT	Filetyp (Zahl)
F$	Filename
LA$	Filelänge in Blocks
BY$	Filelänge in Bytes
RL$	Recordlänge
GF	Gesamtzahl freier Blöcke der Diskette
L$	Bitmuster der belegten Sektoren einer Spur (0 = Sektor belegt, 1 = frei)
P$	aus Directory gelesener Filename in UP »File vorhanden«
FL$	enthält Filelänge in Blocks für UP »Kopf drucken«
I1$	8 Bytes für UP »Bytes ausdrucken«
B$	ASCII-Zeichen der 8 Bytes
AB	Anzahl Blocks pro Seite

Tabelle 3. Die wichtigsten Variablen

```
DIREKTORY

GN  AUFZ.-DICHTE       DOS   ANZAHL BLöCKE    SPUR  SEKTOREN
08  256 BYTES/SEKT.    V2.6  683 (664 FREI)   0018  01-18

Diskettenname: 1541TEST/DEMO   ID: ZX

NR.   FILENAME             BL.   BYTES   1.DB.   TYP
-----------------------------------------------------
001   HOW TO USE           013   03175   17/00   PRG
002   HOW PART TWO         005   01143   17/03   PRG
003   VIC-20 WEDGE         004   00889   17/09   PRG
004   C-64 WEDGE           001   00127   19/00   PRG
005   DOS 5.1              004   00889   19/01   PRG
006   COPY/ALL             011   02667   19/03   PRG
007   PRINTER TEST         009   02159   19/09   PRG
008   DISK ADDR CHANGE     004   00889   16/00   PRG
009   DIR                  004   00889   16/01   PRG
010   VIEW BAM             006   01397   16/03   PRG
011   CHECK DISK           004   00889   16/07   PRG
012   DISPLAY T&S          014   03429   16/15   PRG
013   PERFORMANCE TEST     009   02159   20/02   PRG
014   SEQUENTIAL FILE      005   01143   20/07   PRG
015   RANDOM FILE          013   03175   15/01   PRG

FREIE BLöCKE: 558   IN BYTES: 141732   FREIE EINTRäGE: 129
```

Bild 1. Directoryausdruck

```
BLOCKBELEGUNGSPLAN

Diskettenname: 1541TEST/DEMO   ID: ZX

Sektor                   1 1 1 1 1 1 1 1 1 1 2     FREIE
p    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0     BLöCKE
u
r 1  - - - - - - - - - - - - - - - - - - - - -     21
  2  - - - - - - - - - - - - - - - - - - - - -     21
  3  - - - - - - - - - - - - - - - - - - - - -     21
  4  - - - - - - - - - - - - - - - - - - - - -     21
  5  - - - - - - - - - - - - - - - - - - - - -     21
  6  - - - - - - - - - - - - - - - - - - - - -     21
  7  - - - - - - - - - - - - - - - - - - - - -     21
  8  - - - - - - - - - - - - - - - - - - - - -     21
  9  - - - - - - - - - - - - - - - - - - - - -     21
 10  - - - - - - - - - - - - - - - - - - - - -     21
 11  - - - - - - - - - - - - - - - - - - - - -     21
 12  - - - - - - - - - - - - - - - - - - - - -     21
 13  - - - - - - - - - - - - - - - - - - - - -     21
 14  - - - * - * - - - - - - - * - * - - - - -     17
 15  * * * * * * * * * * * * * * * * * * * * *     00
 16  * * * * * * * * * * * * * * * * * * * * *     00
 17  * * * * * * * * * * * * * * * * * * * * *     00
 18  * * - - * - - - - - - - - - - - - - - / /     16
 19  * * * * * * * * * * * * * * * * * * * / /     00
 20  * * * * * * * * * * * * * * * * * * * / /     00
 21  - - - - - - * - - - - - - - - - - - - / /     18
 22  - - - - - - - - - - - - - - - - - - - / /     19
 23  - - - - - - - - - - - - - - - - - - - / /     19
 24  - - - - - - - - - - - - - - - - - - - / /     19
 25  - - - - - - - - - - - - - - - - - - / / /     18
 26  - - - - - - - - - - - - - - - - - - / / /     18
 27  - - - - - - - - - - - - - - - - - - / / /     18
 28  - - - - - - - - - - - - - - - - - - / / /     18
 29  - - - - - - - - - - - - - - - - - - / / /     18
 30  - - - - - - - - - - - - - - - - - - / / /     18
 31  - - - - - - - - - - - - - - - - - / / / /     17
 32  - - - - - - - - - - - - - - - - - / / / /     17
 33  - - - - - - - - - - - - - - - - - / / / /     17
 34  - - - - - - - - - - - - - - - - - / / / /     17
 35  - - - - - - - - - - - - - - - - - / / / /     17
                                                   ---
                                                   558
```

Bild 2. Blockbelegungsplan einer Diskette

```
Filename:    DIR
Typ:         PRG
Filelänge:   CA. 4 BLOCKS
Spur: 16     Sektor: 2     Block: 3

Byte:
000>   016 012 178 034 034 058 153 034   ..."":."
008>   062 034 059 000 026 006 171 015   >";.....
016>   161 066 036 058 139 066 036 178   .B$:.B$.
024>   034 034 032 167 052 048 049 049   "" .4011
032>   000 053 006 172 015 153 066 036   .5....B$
040>   059 058 139 032 066 036 178 199   ;:. B$..
048>   040 049 051 041 032 167 032 052   (13) . 4
056>   048 050 048 000 073 006 173 015   020.I...
064>   067 036 178 067 036 170 066 036   C$.C$.B$
072>   058 137 032 052 048 049 049 000   :. 4011.
080>   083 006 180 015 152 050 044 067   S....2,C
088>   036 000 093 006 136 019 153 034   $......"
096>   018 034 059 000 126 006 146 019   .";.....
104>   161 035 050 044 065 036 058 153   .#2,A$:.
112>   065 036 059 058 139 032 065 036   A$;:. A$
120>   179 177 199 040 049 051 041 137   ...(13).
128>   053 048 049 048 000 135 006 156   5010....
136>   019 153 034 146 034 000 155 006   ..".."...
144>   016 039 153 032 034 068 045 068   .'. "D-D
152>   073 082 069 067 084 079 082 089   IRECTORY
160>   034 000 178 006 026 039 153 032   ".....'.
168>   034 062 045 068 073 083 075 032   ">-DISK
176>   067 079 077 077 065 078 068 034   COMMAND"
184>   000 201 006 036 039 153 032 034   ...$'. "
192>   081 045 081 085 073 084 032 080   Q-QUIT P
200>   082 079 071 082 065 077 034 000   ROGRAM".
208>   224 006 046 039 153 032 034 083   ...'. "S
216>   045 068 073 083 075 032 083 084   -DISK ST
224>   065 084 085 083 032 034 000 245   ATUS "..
232>   006 116 039 161 065 036 058 139   ..'.A$:.
240>   065 036 178 034 034 167 049 048   A$."".10
248>   049 048 048 000 007 007 216 039   100....'
```

Bild 3. Menüpunkt 2 (Einzelne Blöcke) mit Eingabe Filename/Blocknummer

```
Filename:   DIR
Typ:        PRG
Filelänge:  CA. 4 BLOCKS
Spur: 16    Sektor: 2    Block: 3

Byte:
144>   --- --- --- --- --- 068 045 068    .....D-D
152>   073 082 069 067 084 079 082 089    IRECTORY
160>   034 000 178 006 026 039 153 032    ".....'.
168>   034 062 045 068 073 083 075 032    ">-DISK
176>   067 079 077 077 065 078 068 ---    COMMAND.

192>   081 045 081 085 073 084 032 080    Q-QUIT P
200>   082 079 071 082 065 077 --- ---    ROGRAM..
```

Bild 4. Bytereihen mit Eingabe Filename/Blocknummer

Absatz 1: erstes Byte: 149; letztes Byte: 182;

Absatz 2: erstes Byte: 192; letztes Byte: 205.

```
Filename:   DISPLAY T&S
Typ:        PRG
Filelänge:  CA. 14 BLOCKS

 001 | 002 | 003 | 004 | 005 | 006 | 007 | 008 | 009 | 010
16/15|16/18|15/06|15/16|15/04|15/14|15/02|15/12|15/00|15/10
 011 | 012 | 013 | 014
15/20|15/08|15/18|15/07

GESAMTLÄNGE: 3356 BYTES
```

Bild 5: Blockbelegung von Files. Die obere Spalte enthält die laufende Blocknummer, darunter sind Spur/Sektor angegeben.

```
10 REM    ****************************        <049>
20 REM    ****************************        <059>
30 REM    **                        **        <085>
40 REM    ** JOHANN AUER            **        <074>
50 REM    ** HERMANN-HILLER-STR. 54 **        <028>
60 REM    ** 8263 BURGHAUSEN        **        <058>
70 REM    **                        **        <125>
80 REM    ****************************        <119>
90 REM    ****************************        <129>
100 :                                         <158>
110 :                                         <168>
120 :                                         <178>
130 REM ******************************        <039>
140 REM *******FILE-ORGANISATION*******       <066>
150 REM ******************************        <059>
160 :                                         <218>
170 GOSUB 280:REM ***INITIALISIERUNG****      <048>
180 :                                         <238>
190 GOSUB 460:REM ********MENUE*********      <242>
200 :                                         <002>
210 ON A GOTO 800,990,1380,1710,2160,2900,3460
                          <101>
220 :                                         <022>
230 REM ******************************        <139>
240 REM *******FILE-ORGANISATION*******       <166>
250 REM ******************************        <159>
260 :                                         <063>
270 :                                         <073>
280 REM ********INITIALISIERUNG********       <184>
290 :                                         <093>
300 DIM FT$(3):DIM A$(7):A$(6)="[RIGHT,RVSON]T
    [DOWN,LEFT]T[RVOFF,SPACE,YELLOW]"
    :A$(7)="[RVOFF,SPACE,BLACK]T[LEFT,UP]T[DOWN]"
                          <205>
310 RF=53280:HF=RF+1:A$="":NU$="00000000......."
    :EB=-1:N3=0:RESTORE                       <112>
320 DI$="D"+CHR$(105)+CHR$(115)+CHR$(107)+CHR$(
    101)+CHR$(116)+CHR$(116)                  <063>
330 DI$=DI$+CHR$(101)+CHR$(110)+CHR$(110)+CHR$(
    97)+CHR$(109)++CHR$(101)+":"              <015>
340 FOR I=1 TO 40:READ Q:A$=A$+CHR$(Q):NEXT
                          <161>
350 A$(0)=LEFT$(A$,9):A$(1)=MID$(A$,10,4)
    :A$(2)=MID$(A$,14,10)                     <140>
360 A$(3)=MID$(A$,24,5):A$(4)=MID$(A$,29,7)
    :A$(5)=RIGHT$(A$,5)                       <126>
370 DATA 70,105,108,101,110,97,109,101,58,84,
    121,112,58,70,105                         <031>
380 DATA 108,101,108,123,110,103,101,58,83,112,
    117,114,58,83,101                         <127>
390 DATA 107,116,111,114,58,66,121,116,101,58
                          <243>
400 FT$(0)="REL":FT$(1)="USR":FT$(2)="PRG"
    :FT$(3)="SEQ"                             <200>
410 LZ$="---[SPACE]---[SPACE]---[SPACE]---
    [SPACE]---[SPACE]---[SPACE]---[SPACE10]"
                          <012>
420 PRINT CHR$(8);:POKE 788,52                <212>
430 RETURN                                    <061>
440 :                                         <243>
450 :                                         <253>
460 REM *************MENUE*************       <026>
470 :                                         <017>
480 POKE RF,0:POKE HF,11:PRINT CHR$(14)"[CLEAR,
    DOWN]"TAB(11);                            <250>
490 PRINT"[CYAN]FILE-ORGANISATION"
    :PRINT TAB(11)"[BLACK]UUUUUUUUUUUUUUUUU"
                          <079>
500 PRINT"[RIGHT]IIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
    IIIIIIIF"                                 <030>
510 PRINT A$(6)"GANZES[SPACE]FILE[SPACE,BLACK]
    ....................[SPACE,YELLOW,RVSON]
    1"A$(7)                                   <166>
520 PRINT A$(6)"EINZELNE[SPACE]BLOECKE[SPACE,
    BLACK]...............[SPACE,YELLOW,RVSON]
    2"A$(7)                                   <051>
530 PRINT A$(6)"BYTEREIHEN[SPACE,BLACK]
    .....................[SPACE,YELLOW,RVSON]
    3"A$(7)                                   <114>
540 PRINT A$(6)"BLOCKBELEGUNG[SPACE]VON[SPACE]
    FILES[SPACE,BLACK]........[SPACE,YELLOW,
    RVSON]4"A$(7)                             <210>
550 PRINT A$(6)"DIREKTORY[SPACE,BLACK]
    ....................[SPACE,YELLOW,RVSON]
    5"A$(7)                                   <132>
560 PRINT A$(6)"BAM[SPACE,BLACK]
    ...........................[SPACE,YELLOW,
    RVSON]6"A$(7)                             <182>
570 PRINT A$(6)"ENDE[SPACE,BLACK]
    ...........................[SPACE,YELLOW,
    RVSON]7"A$(7)                             <095>
580 PRINT"[RIGHT,RVSON]T[DOWN,LEFT,RVOFF]T
    [RVSON]TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
    [RVOFF]V[UP,LEFT]T[DOWN]"                 <249>
590 PRINT"[DOWN,SPACE3,CYAN]IHRE[SPACE]WAHL?
    [YELLOW,SPACE]";                          <073>
600 GET EI$:IF EI$<>""THEN 620                <011>
610 PRINT"[SPACE,LEFT]";:FOR T=1 TO 150:NEXT
    :PRINT"T[LEFT]";:FOR T=1 TO 150:NEXT:GOTO 600
                          <224>
620 A=VAL(EI$)::IF A<1 OR A>7 THEN 600        <099>
630 PRINT EI$:FOR T=1 TO 1000:NEXT            <214>
640 IF A=7 OR SP>0 THEN RETURN                <160>
650 PRINT"[UP,SPACE3,YELLOW]AUSGABE[SPACE]AUF
    [SPACE,BLACK]EPSON[YELLOW]-DRUCKER[SPACE,
    BLACK]RX-80[YELLOW]"                      <073>
660 PRINT"[SPACE3]MIT[SPACE,BLACK]DATA-BECKER
    [YELLOW,SPACE]INTERFACE!":FOR T=1 TO 2500
    :NEXT                                     <201>
670 PRINT"[UP2,SPACE3,BLACK]DRUCKER[SPACE]
    EINSCHALTEN[SPACE]UND[SPACE]PAPIER[SPACE4]"
                          <021>
680 PRINT"[SPACE3]POSITIONIEREN!!![SPACE12]"
                          <210>
690 PRINT"[SPACE3,YELLOW]FERTIG[SPACE,BLACK]
    ..................[SPACE,YELLOW,RVSON]F1"
                          <063>
700 GOSUB 5180                                <025>
710 OPEN 1,4,1:POKE 768,61                    <094>
720 PRINT#1,CHR$(7);:CLOSE 1                  <092>
730 POKE 768,139                              <225>
740 IF ST<>-128 THEN 760                      <006>
750 GOTO 700                                  <016>
760 OPEN 4,4,1:GOSUB 3960                     <035>
770 RETURN                                    <147>
780 :                                         <073>
790 :                                         <083>
800 REM*********GANZES FILE***********        <236>
810 :                                         <103>
820 PRINT"[CLEAR,DOWN,SPACE11,BLACK]
    0000000000000000000"                      <145>
830 PRINT"[SPACE11]>>>[WHITE]GANZES[SPACE]FILE
    [BLACK]<<<"                               <009>
840 PRINT"[SPACE11]UUUUUUUUUUUUUUUUUUU"       <096>
850 GOSUB 5230                                <172>
860 AB=0:N=0:BN=1:GOTO 4770                   <112>
870 GOSUB 4050                                <191>
880 AN=0:EN=255                               <255>
890 BP=0:GOSUB 3830:GOSUB 3870                <031>
900 IF S1=0 THEN N=1:EN=S2                    <084>
910 GOSUB 4390:PRINT#4:PRINT#4:GOSUB 4910     <143>
920 IF N=1 THEN N=0:GOTO 940                  <142>
930 SP=S1:SE=S2:BN=BN+1:GOSUB 4100:GOTO 890
                          <094>
```

```
940 GOSUB 3930:PRINT#4:PRINT#4                          <023>
950 IF AB=1 THEN GOSUB 5060                             <169>
960 BN=0:GOTO 190                                       <146>
970 :                                                   <007>
980 :                                                   <017>
990 REM********EINZELNE BLOECKE********                 <095>
1000 :                                                  <037>
1010 PRINT"[CLEAR,SPACE9,BLACK]
   0000000000000000000000"                              <219>
1020 PRINT"[SPACE9]>>>[WHITE]EINZELNE[SPACE]
   BLOECKE[BLACK]<<<"          "                        <080>
1030 PRINT"[SPACE9]TTTTTTTTTTTTTTTTTTTTTT"              <183>
1040 GOSUB 5230                                         <107>
1050 PRINT"[DOWN,SPACE,BLACK]EINGABE:"
   :PRINT"[DOWN,SPACE,CYAN]SPUR/SEKTOR[SPACE,
   BLACK]...........[SPACE,CYAN]1"                      <179>
1060 PRINT"[SPACE]FILENAME/BLOCKNR.[SPACE,BLACK]
   .....[SPACE,CYAN]2"                                  <245>
1070 INPUT"[DOWN,SPACE,BLACK]>>>[SPACE,YELLOW]";
   E$                                                   <152>
1080 E=INT(VAL(E$)):IF E<1 OR E>2 THEN 1100
                         <231>
1090 ON E GOTO 1110,1190                                <095>
1100 PRINT"[UP,RIGHT6,SPACE10,UP2]":GOTO 1070
                         <025>
1110 INPUT"[DOWN,SPACE,BLACK]SPUR/SEKTOR
   :[SPACE,YELLOW]";SP$,SE$                             <215>
1120 SP=INT(VAL(SP$)):SE=INT(VAL(SE$)):I=SP
   :GOSUB 3390                                          <102>
1130 IF(SP=>1 AND SP<=35)AND(SE=>0 AND SE<=MSE)
   THEN 1160                                            <203>
1140 PRINT"[UP]";:FOR T=1 TO 38:PRINT"[SPACE]";
   :FOR T1=1 TO 5:NEXT T1:NEXT T                        <193>
1150 PRINT"[UP2]":GOTO 1110                             <007>
1160 GOSUB 3780:BP=0:GOSUB 3810:BN=0                    <216>
1170 IF A=3 THEN 1460                                   <185>
1180 N2=1:GOTO 1220                                     <139>
1190 N2=0:GOTO 4770                                     <161>
1200 GOTO 4970                                          <017>
1210 SP=S1:SE=S2                                        <160>
1220 GOSUB 3830:GOSUB 3870:AN=0:EN=255
   :IF S1=0 THEN EN=S2                                  <030>
1230 IF N2=1 THEN GOSUB 4100:GOTO 1250                  <068>
1240 GOSUB 4050                                         <050>
1250 N2=0:GOSUB 4390:PRINT#4:PRINT#4:PRINT#4
                         <241>
1260 GOSUB 5060                                         <072>
1270 PRINT"[DOWN,SPACE,BLACK]WEITERER[SPACE]
   BLOCK[SPACE3,YELLOW]J/N?";:GOSUB 5110                <183>
1280 IF A$="J"THEN 1300                                 <160>
1290 GOSUB 3930:BN=0:GOTO 190                           <116>
1300 IF E=1 THEN 1330                                   <058>
1310 PRINT"[DOWN,SPACE,BLACK]GLEICHES[SPACE]
   FILE[SPACE4,YELLOW]J/N?";:GOSUB 5110                 <114>
1320 IF A$="J"THEN 1340                                 <204>
1330 N2=0:GOSUB 3930:GOTO 990                           <148>
1340 N2=1:S1=E1:S2=E2                                   <114>
1350 PRINT"[CLEAR,DOWN2]";:GOTO 1200                    <158>
1360 :                                                  <143>
1370 :                                                  <153>
1380 REM***********BYTEREIHEN**********                 <089>
1390 :                                                  <173>
1400 PRINT"[CLEAR,SPACE12,BLACK]
   0000000000000000"                                    <013>
1410 PRINT"[SPACE12]>>>[WHITE]BYTEREIHEN[BLACK]
   <<<"                                                 <214>
1420 PRINT"[SPACE12]TTTTTTTTTTTTTTTT"                   <238>
1430 GOSUB 5230                                         <242>
1440 N3=0:GOTO 1050                                     <145>
1450 SP=S1:SE=S2:GOSUB 3830                             <038>
1460 INPUT"[DOWN,RIGHT,BLACK]ERSTES[SPACE]BYTE
   :[SPACE2,YELLOW]";EB                                 <228>
1470 IF EB<0 OR EB>255 THEN GOSUB 1510:GOTO 1460
                         <197>
1480 INPUT"[DOWN,RIGHT,BLACK]LETZTES[SPACE]BYTE
   :[SPACE,YELLOW]";LB                                  <084>
1490 IF LB<0 OR LB>255 OR LB<EB THEN GOSUB 1510
   :GOTO 1480                                           <097>
1500 GOTO 1520                                          <050>
1510 PRINT"[UP]";TAB(14);"[SPACE10,UP2]":RETURN
                         <046>
1520 IF N3<>0 THEN GOTO 1670                            <147>
1530 IF E=1 THEN GOSUB 4100:GOTO 1550                   <057>
1540 GOSUB 4050                                         <096>
1550 AN=EB:EN=LB:GOSUB 4390                             <128>
1560 PRINT"[CLEAR,DOWN]";:GOSUB 5060                    <108>
1570 PRINT"[DOWN,RIGHT,BLACK]WEITERE[SPACE]
   REIHE[SPACE4,YELLOW]J/N?";:GOSUB 5110                <178>
1580 IF A$="J"THEN 1600                                 <208>
1590 GOSUB 3930:PRINT#4:PRINT#4:PRINT#4:EB=-1
   :N3=0:GOTO 190                                       <243>
1600 PRINT"[DOWN,RIGHT,BLACK]GLEICHER[SPACE]
   BLOCK[SPACE3,YELLOW]J/N?";:GOSUB 5110                <252>
1610 IF A$="J"THEN N3=1:PRINT"[DOWN]";:GOTO 1460
                         <124>
1620 PRINT#4:PRINT#4                                    <044>
1630 IF E=1 THEN 1660                                   <139>
1640 PRINT"[DOWN,RIGHT,BLACK]GLEICHES[SPACE]
   FILE[SPACE4,YELLOW]J/N?";:GOSUB 5110                 <218>
1650 IF A$="J"THEN N3=2:S1=E1:S2=E2
   :PRINT"[DOWN]";:GOTO 4970                            <124>
1660 GOSUB 3930:PRINT#4:GOTO 1380                       <115>
1670 IF N3=1 THEN 1550                                  <237>
1680 GOSUB 4100:GOTO 1550                               <118>
1690 :                                                  <218>
1700 :                                                  <228>
1710 REM*********BLOCKBELEGUNG*********                 <235>
1720 :                                                  <248>
1730 PRINT"[CLEAR,SPACE10,BLACK]
   00000000000000000000"                                <131>
1740 PRINT"[SPACE10]>>>[WHITE]BLOCKBELEGUNG
   [BLACK]<<<"                                          <230>
1750 PRINT"[SPACE10]TTTTTTTTTTTTTTTTTTTT"               <097>
1760 GOSUB 5230:GOTO 4770                               <210>
1770 PRINT"[DOWN2,RIGHT]FORMAT
   :[SPACE2,WHITE]BLOCKNUMMER"
   :PRINT"[SPACE,BLACK]YYYYYY[SPACE3]
   ***********"                                         <004>
1780 PRINT"[SPACE10,WHITE]SPUR/SEKTOR"                  <045>
1790 GOSUB 4050:PRINT#4:RE=10:RD=295:X1=39:X2=1
                         <238>
1800 AR=INT(FL/10+0.95):GOSUB 4300                      <254>
1810 FOR I=1 TO AR:PRINT#4,"[SPACE6]";                  <174>
1820 IF FL/10=INT(FL/10)THEN 1840                       <026>
1830 IF I=AR THEN GOSUB 2120                            <063>
1840 J=0                                                <099>
1850 J=J+1:SU=(I-1)*10+J:BN$=STR$(SU)                   <202>
1860 L3=LEN(BN$)-1:BN$="[SPACE]"+LEFT$(NU$,
   3-L3)+RIGHT$(BN$,L3)+"[SPACE]"                       <040>
1870 PRINT#4,BN$;                                       <060>
1880 IF J+0.5=>RE THEN 1920                             <222>
1890 PRINT#4,CHR$(27);"K";CHR$(5);CHR$(0);
   :A(3)=254                                            <062>
1900 FOR IJ=1 TO 5:PRINT#4,CHR$(A(IJ));:NEXT IJ
                         <060>
1910 GOTO 1850                                          <212>
1920 C$="3":C3=20:GOSUB 4030:PRINT#4                    <111>
1930 PRINT#4,"[SPACE6]";:PRINT#4,CHR$(27);"K";
   CHR$(X1);CHR$(X2);                                   <121>
1940 FOR IJ=1 TO RD:PRINT#4,CHR$(64);:NEXT IJ
   :C3=8:GOSUB 4030:PRINT#4                             <056>
1950 C3=40:GOSUB 4030:PRINT#4,"[SPACE6]";               <112>
1960 J=0:SS=0                                           <157>
1970 J=J+1:SP$=STR$(S1):SE$=STR$(S2)
   :IF S1=0 THEN SS$="[SPACE]SSB[SPACE]":SS=SS+1
   :GOTO 2000                                           <122>
1980 L4=LEN(SP$)-1:L5=LEN(SE$)-1                        <197>
1990 SS$=LEFT$(NU$,2-L4)+RIGHT$(SP$,
   L4)+"/"+LEFT$(NU$,2-L5)+RIGHT$(SE$,L5)               <071>
2000 PRINT#4,SS$;                                       <212>
2010 IF J+0.5=>RE THEN 2040                             <090>
2020 PRINT#4,CHR$(27);"K";CHR$(5);CHR$(0);
   :A(3)=255                                            <193>
2030 FOR IJ=1 TO 5:PRINT#4,CHR$(A(IJ));:NEXT IJ
   :IF S1=0 THEN 1970                                   <097>
2040 IF S1=0 THEN 2060                                  <095>
2050 SP=S1:SE=S2:BP=0:GOSUB 3830:GOSUB 3870
   :IF J<RE THEN 1970                                   <153>
2060 PRINT#4:NEXT I:PRINT#4:GOSUB 4320
   :GOSUB 3930                                          <017>
2070 GOSUB 4220:PRINT#4,"GESAMTL"CHR$(91)"NGE:";
   :GOSUB 4270                                          <159>
2080 PRINT#4,(FL-1-SS)*254+S2"BYTES":PRINT#4
   :PRINT#4:PRINT#4:PRINT#4                             <105>
2090 PRINT"[DOWN]";:GOSUB 5060
   :PRINT"[DOWN,RIGHT,BLACK]WEITERES[SPACE]FILE
   [SPACE4,YELLOW]J/N?";:GOSUB 5110                     <183>
2100 IF A$="J"THEN 1710                                 <220>
2110 GOTO 190                                           <105>
```

```
2120 RE=10*(FL/10-INT(FL/10))
     :RD=INT(RE*29.5+0.5)                       <190>
2130 X1=RD-INT(RD/256)*256:X2=INT(RD/256):RETURN
                    <041>
2140 :                                          <158>
2150 :                                          <168>
2160 REM***********DIREKTORY***********         <096>
2170 :                                          <188>
2180 PRINT"[CLEAR,SPACE12,BLACK]OOOOOOOOOOOOOOO"
                    <099>
2190 PRINT"[SPACE12]>>>[WHITE]DIREKTORY[BLACK]
     <<<"                                       <179>
2200 PRINT"[SPACE12]UUUUUUUUUUUUUUU":GOSUB 5230
     :GOSUB 5280                                <108>
2210 PRINT"[DOWN,SPACE,BLACK]ABKUERZUNGEN:"
     :PRINT"[SPACE,YELLOW]BL.[SPACE3,BLACK]-
     [SPACE,CYAN]FILELAENGE[SPACE]IN[SPACE]
     BLOECKEN;"                                 <161>
2220 PRINT"[SPACE,YELLOW]BYTES[SPACE,BLACK]-
     [SPACE,CYAN]UNGEFAEHRE[SPACE]FILELAENGE
     [SPACE]IN[SPACE16]BYTES[SPACE]";           <145>
2230 PRINT"(BL*254)":PRINT"[SPACE,YELLOW]1.DB.
     [SPACE,BLACK]-[SPACE,CYAN]1.[SPACE]
     DATENBLOCK[SPACE]DES[SPACE]FILES;"         <094>
2240 PRINT"[SPACE,YELLOW]TYP[SPACE3,BLACK]-
     [SPACE,CYAN]FILETYP.":PRINT"[DOWN,SPACE,
     BLACK]NUR[SPACE]BEI[SPACE]RELATIVEN[SPACE]
     DATEIEN:"                                  <066>
2250 PRINT"[SPACE,YELLOW]1.SSB[SPACE,BLACK]-
     [SPACE,CYAN]1.SIDE-SEKTOR-BLOCK;"          <076>
2260 PRINT"[SPACE,YELLOW]RL[SPACE4,BLACK]-
     [SPACE,CYAN]RECORDLAENGE."                 <234>
2270 GOSUB 3780:PRINT#4,"[SPACE5]";             <246>
2280 GOSUB 4240:C$="W":C3=1:GOSUB 4030          <086>
2290 PRINT#4,"DIREKTORY":PRINT#4:C3=0:GOSUB 4030
     :GOSUB 4270                                <173>
2300 GOSUB 4300:PRINT#4,"[SPACE6]GN[SPACE2]
     AUFZ.-DICHTE[SPACE5]DOS[SPACE3]";          <198>
2310 PRINT#4,"ANZAHL[SPACE]BL"CHR$(92)"CKE
     [SPACE3]SPUR[SPACE2]SEKT."                 <224>
2320 PRINT#4,"[SPACE6]08[SPACE2]256[SPACE]
     BYTES/SEKT.[SPACE2]V2.6[SPACE2]683[SPACE]
     (664[SPACE]FREI)[SPACE2]0018[SPACE2]01-18"
                    <217>
2330 PRINT#4                                    <239>
2340 GOSUB 5320:GOSUB 3930:GOSUB 4320           <170>
2350 PRINT#4,"[SPACE5]";:GOSUB 4240:PRINT#4,DI$;
     :GOSUB 4270:PRINT#4,DN$;                   <020>
2360 GOSUB 4240:PRINT#4,"ID:";:GOSUB 4270       <164>
2370 PRINT#4,ID$:PRINT#4                        <250>
2380 PRINT#4,"[SPACE5]NR.[SPACE2]FILENAME
     [SPACE10]BL.[SPACE2]BYTES";                <030>
2390 PRINT#4,"[SPACE2]1.DB.[SPACE3]TYP"
     :GOSUB 4340:PRINT#4,"[SPACE5]";            <238>
2400 FOR I=1 TO 46:PRINT#4,"-";:NEXT I:PRINT#4
                    <013>
2410 ET$="":I=0:FB=0:SP=18:SE=1:BP=0:GOSUB 3810
                    <029>
2420 GOSUB 3870:BP=2:GOSUB 3840                 <034>
2430 FOR J=1 TO 8:I=I+1                         <106>
2440 FOR J1=1 TO 32:GET#3,A$
     :IF A$=""THEN A$=CHR$(0)                   <097>
2450 ET$=ET$+A$:NEXT J1:ET$=LEFT$(ET$,30)       <027>
2460 FT=ASC(LEFT$(ET$,1)):F1=FT
     :IF FT=0 THEN I=I-1:GOTO 2650              <208>
2470 EN$=LEFT$(NU$,4-LEN(STR$(I)))+RIGHT$(STR$(
     I),LEN(STR$(I))-1)                         <017>
2480 IF FT<128 THEN OF$="*"                     <172>
2490 OF$="[SPACE]"                              <114>
2500 FT=(F1 OR 128)AND 135:FT$=OF$+FT$(132-FT)
                    <107>
2510 AS=ASC(MID$(ET$,2,1)):SP$=LEFT$(NU$,
     3-LEN(STR$(AS)))                           <033>
2520 SP$=SP$+RIGHT$(STR$(AS),LEN(STR$(AS))-1)
                    <083>
2530 AS=ASC(MID$(ET$,3,1)):SE$=LEFT$(NU$,
     3-LEN(STR$(AS)))                           <043>
2540 SE$=SE$+RIGHT$(STR$(AS),LEN(STR$(AS))-1)
                    <081>
2550 DB$=SP$+"/"+SE$                            <165>
2560 F$=MID$(ET$,4,16)                          <241>
2570 AS=ASC(RIGHT$(ET$,1))*256+ASC(MID$(ET$,29,
     1)):FB=FB+AS                               <181>
2580 LA$=LEFT$(NU$,4-LEN(STR$(AS)))+RIGHT$(STR$
     (AS),LEN(STR$(AS))-1)                      <091>
2590 AS=(AS*256)-(AS*2)-127                     <079>
2600 BY$=LEFT$(NU$,6-LEN(STR$(AS)))+RIGHT$(STR$
     (AS),LEN(STR$(AS))-1)                      <127>
2610 PRINT#4,"[SPACE5]"EN$"[SPACE2]"F$"[SPACE2]
     "LA$;                                      <013>
2620 PRINT#4,"[SPACE2]"BY$"[SPACE2]"DB$"[SPACE2]
     "FT$;                                      <108>
2630 IF RIGHT$(FT$,3)="REL"THEN GOSUB 2780      <240>
2640 PRINT#4                                    <038>
2650 ET$="":NEXT J                              <029>
2660 IF S1=0 THEN 2680                          <214>
2670 SP=S1:SE=S2:BP=0:GOSUB 3830:GOTO 2420      <040>
2680 FB=664-FB:GOSUB 4360                       <035>
2690 PRINT#4:GOSUB 4300:PRINT#4,"[SPACE6]";
                    <151>
2700 PRINT#4,"FREIE[SPACE]BL"CHR$(92)"CKE
     :"FB"[SPACE2]IN[SPACE]BYTES:"(FB*254);     <030>
2710 PRINT#4,"[SPACE2]FREIE[SPACE]
     EINTR"CHR$(91)"GE:"144-I:PRINT#4:PRINT#4
     :PRINT#4:PRINT#4                           <250>
2720 GOSUB 4320:GOSUB 3930                      <150>
2730 GOSUB 5060                                 <012>
2740 PRINT"[DOWN,RIGHT,BLACK]WEITERE[SPACE]
     DISKETTE[SPACE,YELLOW]J/N?";               <170>
2750 GOSUB 5110                                 <028>
2760 IF A$="J"THEN 2160                         <114>
2770 GOTO 190                                   <255>
2780 AS=ASC(MID$(ET$,20,1)):SP$=LEFT$(NU$,
     3-LEN(STR$(AS)))                           <096>
2790 SP$=SP$+RIGHT$(STR$(AS),LEN(STR$(AS))-1)
                    <098>
2800 AS=ASC(MID$(ET$,21,1)):SE$=LEFT$(NU$,
     3-LEN(STR$(AS)))                           <106>
2810 SE$=SE$+RIGHT$(STR$(AS),LEN(STR$(AS))-1)
                    <096>
2820 DB$="[SPACE]"+SP$+"/"+SE$                  <163>
2830 AS=ASC(MID$(ET$,22,1)):RL$=LEFT$(NU$,
     4-LEN(STR$(AS)))                           <145>
2840 RL$="[SPACE]"+RL$+RIGHT$(STR$(AS),
     LEN(STR$(AS))-1)                           <121>
2850 GOSUB 4300:PRINT#4,"[SPACE3]1.SSB:";
     :GOSUB 4320:PRINT#4,DB$;                   <090>
2860 GOSUB 4300:PRINT#4,"[SPACE3]R"CHR$(108)":";
     :GOSUB 4320:PRINT#4,RL$;                   <124>
2870 RETURN                                     <207>
2880 :                                          <133>
2890 :                                          <143>
2900 REM*************BAM*************           <086>
2910 :                                          <163>
2920 PRINT"[CLEAR,SPACE14,BLACK]OOOOOOOOOOO"
                    <102>
2930 PRINT"[SPACE14]>>>[WHITE]B[SPACE]A[SPACE]M
     [BLACK]<<<":PRINT"[SPACE14]UUUUUUUUUUU"    <172>
2940 PRINT"[SPACE9,YELLOW](BLACK]B[YELLOW]LOCK
     [SPACE,BLACK]A[YELLOW]VAILABLE[SPACE,BLACK]M
     [YELLOW]AP)"                               <039>
2950 PRINT"[DOWN2,SPACE,BLACK]FORMAT
     :[SPACE,CYAN]HORIZONTAL[SPACE,BLACK]-[SPACE,
     CYAN]SEKTOREN"                             <019>
2960 PRINT"[SPACE9]VERTIKAL[SPACE3,BLACK]-
     [SPACE,CYAN]SPUREN"                        <147>
2970 PRINT"[DOWN,RIGHT,BLACK]ZEICHENBEDEUTUNG
     :[SPACE,CYAN]'*'[BLACK]=[CYAN]BLOCK[SPACE]
     BELEGT;"                                   <127>
2980 PRINT TAB(19)"[CYAN]'-'[BLACK]=[CYAN]BLOCK
     [SPACE]FREI;"                              <148>
2990 PRINT TAB(19)"[CYAN]'/'[BLACK]=[CYAN]BLOCK
     [SPACE]NICHT[SPACE]VOR-":PRINT TAB(23)"HANDE
     N."                                        <253>
3000 GOSUB 5230:GOSUB 5280                      <176>
3010 T1$="":L$="":GOSUB 3780:GOSUB 5320
     :PRINT#4,"[SPACE5]";                       <231>
3020 GOSUB 4240:C$="W":C3=1:GOSUB 4030          <061>
3030 PRINT#4,"BLOCKBELEGUNGSPLAN":PRINT#4:C3=0
     :GOSUB 4030:GOSUB 4270                     <009>
3040 C$="3":C3=24:GOSUB 4030:PRINT#4,"[SPACE5]";
     :GOSUB 4240:PRINT#4,DI$;                   <049>
3050 GOSUB 4270:PRINT#4,DN$;:GOSUB 4240
     :PRINT#4,"ID:";:GOSUB 4270                 <015>
3060 PRINT#4,ID$                                <168>
3070 PRINT#4:PRINT#4,"[SPACE5]
     S"CHR$(101)CHR$(107)CHR$(116)CHR$(111)CHR$(1
     14);                                       <060>
3080 PRINT#4,"[SPACE17]1[SPACE]1[SPACE]1[SPACE]
     1[SPACE]1[SPACE]1[SPACE]1[SPACE]1[SPACE]1
```

```
[SPACE]1[SPACE]2[SPACE3]FREIE"                    <215>
3090 PRINT#4,"[SPACE5]"CHR$(112)"[SPACE2]0
   [SPACE]1[SPACE]2[SPACE]3[SPACE]4[SPACE]5
   [SPACE]6[SPACE]7[SPACE]8[SPACE]9[SPACE]0
   [SPACE]1[SPACE]2[SPACE]3[SPACE]4[SPACE]5
   [SPACE]6[SPACE]7[SPACE]8[SPACE]9[SPACE]0
   [SPACE3]";                                      <207>
3100 PRINT#4,"BL"CHR$(92)"CKE"
   :PRINT#4,"[SPACE5]"CHR$(117)                    <179>
3110 GF=0:BP=4:GOSUB 3840                          <233>
3120 FOR I=1 TO 35:IF I=1 THEN 3380                <091>
3130 GOSUB 3390:IF I<10 THEN PRINT#4,"[SPACE5]
   "I;:GOTO 3150                                   <183>
3140 PRINT#4,"[SPACE4]"I;                          <016>
3150 FOR J=1 TO 4:GET#3,A$:IF A$=""THEN A$=CHR$
   (0)                                             <200>
3160 AC=ASC(A$):IF J=1 THEN FB=AC:GF=GF+FB
   :GOTO 3210                                      <046>
3170 T1=2*(AC/2-INT(AC/2)):T1$=RIGHT$(STR$(T1),
   1)                                              <241>
3180 L$=L$+T1$:AC=INT(AC/2):IF AC=0 THEN 3200
                        <205>
3190 GOTO 3170                                     <214>
3200 L$=L$+LEFT$(NU$,8-(LEN(L$)-(J-2)*8))
   :GOTO 3220                                      <029>
3210 IF I=18 THEN GF=GF-FB                         <043>
3220 NEXT J                                        <108>
3230 FOR J=1 TO 21:IF J<=MSE THEN 3250             <041>
3240 PRINT#4,"/[SPACE]";:GOTO 3270                 <233>
3250 IF MID$(L$,J,1)="0"THEN PRINT#4,"*[SPACE]";
   :GOTO 3270                                      <164>
3260 PRINT#4,"-[SPACE]";                           <108>
3270 NEXT J:IF FB<10 THEN PRINT#4,"[SPACE2]
   0"RIGHT$(STR$(FB),1):GOTO 3290                  <183>
3280 PRINT#4,"[SPACE]"FB                           <160>
3290 L$="":NEXT I                                  <081>
3300 FOR I=1 TO 51:PRINT#4,"[SPACE]";:NEXT I
   :IF GF>99 THEN PRINT#4,"---":GOTO 3320          <199>
3310 PRINT#4,"[SPACE]--"                           <144>
3320 FOR I=1 TO 54-LEN(STR$(GF))
   :PRINT#4,"[SPACE]";:NEXT I:PRINT#4,GF:GF=0
                        <061>
3330 GOSUB 4360:PRINT#4:PRINT#4:PRINT#4            <123>
3340 GOSUB 3930:GOSUB 5060                         <007>
3350 PRINT"[DOWN,RIGHT,BLACK]WEITERE[SPACE]
   DISKETTE[SPACE,YELLOW]J/N?";:GOSUB 5110         <157>
3360 IF A$="J"THEN 2900                            <175>
3370 GOTO 190                                      <090>
3380 MSE=21:PRINT#4,"[SPACE5]"CHR$(114)"1[SPACE]
   ";:GOTO 3150                                    <155>
3390 IF I=>1 AND I<=17 THEN MSE=21:GOTO 3430
                        <166>
3400 IF I=>18 AND I<=24 THEN MSE=19:GOTO 3430
                        <237>
3410 IF I=>25 AND I<=30 THEN MSE=18:GOTO 3430
                        <241>
3420 IF I=>31 AND I<=35 THEN MSE=17                <111>
3430 RETURN                                        <001>
3440 :                                             <183>
3450 :                                             <193>
3460 REM*************ENDE**************            <170>
3470 :                                             <213>
3480 PRINT"[CLEAR,GREEN]"CHR$(9);:CLOSE 4
   :POKE HF,0                                      <136>
3490 POKE 788,49:CLR:GOTO 5430                     <235>
3500 :                                             <243>
3510 :                                             <253>
3520 REM*******UP FILE VORHANDEN*******            <018>
3530 SP=18:SE=1:BP=0:P$=""                         <156>
3540 GOSUB 3780                                    <064>
3550 GOSUB 3810:GOSUB 3870                         <221>
3560 BP=2:GOSUB 3840                               <001>
3570 GET#3,A$:IF A$=""THEN A$=CHR$(0)              <173>
3580 IF ASC(A$)=0 THEN 3660                        <104>
3590 GET#3,A$,A$                                   <045>
3600 FOR I=1 TO 16                                 <214>
3610 GET#3,A$:IF A$=""THEN A$=CHR$(0)              <214>
3620 IF A$=CHR$(160)THEN 3640                      <247>
3630 P$=P$+A$:NEXT                                 <161>
3640 PRINT"[UP,RIGHT11,SPACE18]"
   :PRINT"[UP,RIGHT11]"P$                          <078>
3650 IF F$=P$THEN P$="":GOTO 3700                  <009>
3660 BP=BP+32:IF BP>226 THEN P$="":GOTO 3680
                        <134>
3670 GOSUB 3840:P$="":GOTO 3570                    <246>
3680 IF S1=0 THEN GOSUB 3930:GOTO 4840             <245>
3690 GOSUB 3930:SP=S1:SE=S2:BP=0:GOTO 3550         <046>
3700 BP=BP+1:GOSUB 3840                            <201>
3710 GOSUB 3870:SP=S1:SE=S2                        <007>
3720 BP=BP-1:GOSUB 3840:GOSUB 3870
   :FT=(S1 OR 128)AND 135                          <101>
3730 FT$=FT$(132-FT)                               <250>
3740 BP=BP+28:GOSUB 3840:GOSUB 3870:FL=S1+256*S2
                        <061>
3750 FL$="CA."+STR$(FL)+"[SPACE]BLOCKS":S1=SP
   :S2=SE                                          <043>
3760 RETURN                                        <076>
3770 :                                             <002>
3780 REM*****FLOPPY INITIALISIEREN*****            <248>
3790 OPEN 1,8,15,"I":CLOSE 1:RETURN                <046>
3800 :                                             <032>
3810 REM************B-R/B-P************            <030>
3820 OPEN 2,8,15:OPEN 3,8,2,"#"                    <034>
3830 PRINT#2,"U1[SPACE]2[SPACE]0[SPACE]";SP;SE
                        <215>
3840 PRINT#2,"B-P[SPACE]2[SPACE]";BP               <007>
3850 RETURN                                        <167>
3860 :                                             <093>
3870 REM*********2 BYTES HOLEN*********            <223>
3880 GET#3,A1$,A2$:IF A1$=""THEN A1$=CHR$(0)
                        <059>
3890 IF A2$=""THEN A2$=CHR$(0)                     <145>
3900 S1=ASC(A1$):S2=ASC(A2$):ND=S1*S2              <128>
3910 RETURN                                        <227>
3920 :                                             <153>
3930 REM******KANAELE SCHLIESSEN*******            <252>
3940 CLOSE 3:CLOSE 2:RETURN                        <026>
3950 :                                             <183>
3960 REM****DRUCKER INITIALISIEREN*****            <185>
3970 PRINT#4,CHR$(27);CHR$(64);                    <002>
3980 PRINT#4,CHR$(27);CHR$(115);CHR$(1);           <189>
3990 RETURN                                        <051>
4000 :                                             <233>
4010 REM*******SCHRIFTART SENDEN*******            <075>
4020 PRINT#4,CHR$(C1);CHR$(C2);CHR$(C3);:RETURN
                        <219>
4030 PRINT#4,CHR$(27);C$;CHR$(C3);:RETURN          <180>
4040 :                                             <017>
4050 REM*********KOPF DRUCKEN**********            <202>
4060 GOSUB 4220:PRINT#4,A$(0);:GOSUB 4270
   :PRINT#4,"[SPACE3]";F$                          <066>
4070 GOSUB 4220:PRINT#4,A$(1);:GOSUB 4270
   :PRINT#4,"[SPACE8]";FT$                         <161>
4080 GOSUB 4220:PRINT#4,A$(2);:GOSUB 4270
   :PRINT#4,"[SPACE2]";FL$                         <164>
4090 IF A=4 THEN 4190                              <048>
4100 GOSUB 4220:PRINT#4,A$(3);:GOSUB 4270
   :PRINT#4,SP;                                    <099>
4110 IF SP<10 THEN PRINT#4,"[SPACE]";              <126>
4120 PRINT#4,"[SPACE3]";:GOSUB 4240
   :PRINT#4,A$(4);:GOSUB 4270:PRINT#4,SE;          <032>
4130 IF BN>0 THEN PRINT#4,"[SPACE3]";:GOTO 4150
                        <217>
4140 PRINT#4,"[SPACE]":GOTO 4180                   <008>
4150 IF SE<10 THEN PRINT#4,"[SPACE]";              <155>
4160 GOSUB 4240:PRINT#4,"B"CHR$(108)CHR$(111)CH
   R$(99)CHR$(107)":";                             <174>
4170 GOSUB 4270:PRINT#4,BN                         <118>
4180 GOSUB 4250:PRINT#4:PRINT#4,"[SPACE5]";A$(5)
                        <094>
4190 GOSUB 4280                                    <201>
4200 RETURN                                        <006>
4210 :                                             <188>
4220 REM*****SCHRIFTART AUSWAEHLEN*****            <166>
4230 PRINT#4,"[SPACE5]";                           <013>
4240 C$="-":C3=1:GOSUB 4030                        <075>
4250 C$="E":C3=0:GOSUB 4030                        <108>
4260 RETURN                                        <066>
4270 C$="-":GOSUB 4030                             <214>
4280 C$="F":GOSUB 4030                             <249>
4290 RETURN                                        <096>
4300 C$="M":C3=0:GOSUB 4030                        <166>
4310 RETURN                                        <116>
4320 C$="P":C3=0:GOSUB 4030                        <189>
4330 RETURN                                        <136>
4340 C$="0":C3=0:GOSUB 4030                        <177>
4350 RETURN                                        <156>
4360 C$="2":C3=0:GOSUB 4030                        <200>
4370 RETURN                                        <177>
4380 :                                             <103>
```

```
4390 REM*******BYTES AUSDRUCKEN********       <184>
4400 GOSUB 4340                               <153>
4410 LZ=32*(AN/8-INT(AN/8)):L1$=LEFT$(LZ$,LZ)
   :IF EB=>0 THEN AN=INT(EB/8)*8              <253>
4420 LZ=32-32*((EN+1)/8-INT((EN+1)/8))
   :IF LZ=32 THEN LZ=0                        <201>
4430 L2$=LEFT$(LZ$,LZ)                        <104>
4440 BP=AN:GOSUB 3840                         <210>
4450 FOR J=AN TO EN STEP 8:H$=STR$(J):L1=LEN(H$)
                        <055>
4460 H$="[SPACE6]"+LEFT$(NU$,4-L1)+RIGHT$(H$,
   L1-1)+CHR$(62)+"[SPACE3]"                  <190>
4470 GOSUB 4300                               <219>
4480 PRINT#4,H$;L1$;:L1$="":I1$=""            <171>
4490 FOR I=0 TO 7                             <034>
4500 GET#3,A$:IF A$=""THEN A$=CHR$(0)         <083>
4510 IF EB=>0 AND J+I<EB THEN A$=".":GOSUB 4730
   :GOTO 4560                                 <209>
4520 AS=ASC(A$):I$=STR$(AS):L2=LEN(I$)        <104>
4530 I$=LEFT$(NU$,4-L2)+RIGHT$(I$,L2-1)+"[SPACE]
   ":I1$=I1$+I$                               <047>
4540 GOSUB 4650                               <041>
4550 IF J+I=EN THEN 4570                      <091>
4560 NEXT I:I=I-1                             <006>
4570 IF J+I=EN THEN I1$=I1$+L2$:J=EN:L2$=""
                        <116>
4580 I1$=I1$+"[SPACE2]"                       <209>
4590 PRINT#4,I1$;:GOSUB 4320                  <096>
4600 B$=LEFT$(B$,LEN(B$))+RIGHT$(NU$,8-LEN(B$))
                        <090>
4610 PRINT#4,B$:B$=""                         <008>
4620 NEXT J                                   <234>
4630 GOSUB 4360:PRINT#4:RETURN                <080>
4640 :                                        <076>
4650 REM***OPERAT.IN ZEICHEN UMWAND.***       <087>
4660 IF AS<32 OR AS>90 THEN A$=".":GOTO 4750
                        <158>
4670 IF AS=170 THEN A$="+":GOTO 4750          <121>
4680 IF AS=171 THEN A$="-":GOTO 4750          <134>
4690 IF AS=173 THEN A$="/":GOTO 4750          <148>
4700 IF AS=172 THEN A$="*":GOTO 4750          <152>
4710 IF AS=173 THEN A$="/":GOTO 4750          <168>
4720 IF AS=177 THEN A$=">":GOTO 4750          <197>
4730 IF AS=178 THEN A$="=":GOTO 4750          <207>
4740 IF AS=179 THEN A$="<"                    <070>
4750 B$=B$+A$:RETURN                          <245>
4760 :                                        <228>
4770 REM*******EINGABE FILENAME********       <229>
4780 INPUT"[DOWN,SPACE,BLACK]FILENAME
   :[SPACE,YELLOW]";F$                        <102>
4790 IF LEN(F$)<=16 THEN 4820                 <018>
4800 PRINT"[UP,SPACE13]";                     <123>
4810 FOR T=1 TO LEN(F$):PRINT"[SPACE]";:NEXT T
   :PRINT"[UP2]":GOTO 4780                    <231>
4820 GOSUB 3520:E1=S1:E2=S2                   <011>
4830 ON A GOTO 870,1200,1200,1770             <195>
4840 PRINT"[DOWN2,SPACE,WHITE]
   FILE NICHT VORHANDEN!"                     <154>
4850 PRINT"[DOWN,SPACE]F1[BLACK]=NEUE[SPACE]
   EINGABE[SPACE2,WHITE]F3[BLACK]=MENUE"      <147>
4860 GET A$                                   <020>
4870 IF A$=CHR$(133)THEN 210                  <164>
4880 IF A$=CHR$(134)THEN 190                  <182>
4890 GOTO 4860                                <136>
4900 :                                        <113>
4910 REM******2 BLOCKS PRO SEITE*******       <077>
4920 AB=AB+1:IF AB=2 THEN 4940                <130>
4930 RETURN                                   <227>
4940 AB=0:PRINT#4,CHR$(12):GOSUB 3960         <010>
4950 RETURN                                   <247>
4960 :                                        <173>
4970 REM******EINGABE BLOCKNUMMER******       <046>
4980 PRINT"[DOWN,RIGHT,WHITE]FILE[SPACE]BELEGT";
   FL"[SPACE]BLOECKE!"                        <145>
4990 INPUT"[DOWN,RIGHT,BLACK]BLOCKNUMMER
   :[SPACE2,YELLOW]";BN:IF BN<1 OR BN>FL THEN G
   OSUB 3930:GOTO 210                         <143>
5000 N1=1                                     <253>
5010 IF N1=BN THEN 5040                       <097>
5020 SP=S1:SE=S2:BP=0:GOSUB 3830:GOSUB 3870
                        <109>
5030 N1=N1+1:GOTO 5010                        <205>
5040 ON A GOTO 0,1210,1450                    <052>
5050 :                                        <007>
5060 REM******SEITENVORSCHUB J/N*******       <131>
5070 PRINT"[DOWN,RIGHT,BLACK]SEITENVORSCHUB
   [SPACE3,YELLOW]J/N?";:GOSUB 5110           <157>
5080 IF A$="J"THEN GOSUB 4940                 <032>
5090 RETURN                                   <131>
5100 :                                        <057>
5110 REM********TASTATURABFRAGE********       <152>
5120 GET A$                                   <026>
5130 IF A$="J"THEN PRINT"[LEFT4,CYAN]J"
   :GOTO 5160                                 <062>
5140 IF A$="N"THEN PRINT"[LEFT2,CYAN]N"
   :GOTO 5160                                 <022>
5150 GOTO 5120                                <131>
5160 FOR T=1 TO 1000:NEXT T:RETURN            <049>
5170 :                                        <128>
5180 REM*********F1 - ABFRAGE**********       <137>
5190 GET A$                                   <096>
5200 IF A$=CHR$(133)THEN RETURN               <234>
5210 GOTO 5190                                <198>
5220 :                                        <178>
5230 REM*******DISKETTE EINLEGEN*******       <001>
5240 PRINT"[DOWN,SPACE,YELLOW]ZU[SPACE]
   BEARBEITENDE[SPACE]DISKETTE[SPACE]EINLEGEN!"
                        <230>
5250 IF A=5 OR A=6 THEN RETURN                <087>
5260 FOR T=1 TO 1000:NEXT T:RETURN            <149>
5270 :                                        <228>
5280 REM*******FERTIG .......F1********       <051>
5290 PRINT"[SPACE,YELLOW]FERTIG[SPACE,BLACK]
   ..........................[SPACE,YELLOW]F1"
                        <239>
5300 GOSUB 5180:RETURN                        <235>
5310 :                                        <012>
5320 REM****DISK.-NAME UND ID LESEN****       <077>
5330 SP=18:SE=0:BP=144:GOSUB 3810             <002>
5340 D$="":J=0                                <180>
5350 FOR I=1 TO 20:GET#3,A$:IF A$=""THEN A$=CHR
   $(0)                                       <149>
5360 IF J>0 THEN 5380                         <049>
5370 IF A$=CHR$(160)THEN J=I-1:GOTO 5390      <187>
5380 D$=D$+A$                                 <170>
5390 NEXT I                                   <238>
5400 DN$="[SPACE]"+LEFT$(D$,J)+"[SPACE2]"
   :ID$="[SPACE]"+RIGHT$(D$,2)                <211>
5410 RETURN                                   <197>
5420 :                                        <123>
5430 END                                      <203>
```

...Compüter
Sind schließlich
aüch nür Menschen!

NEU

HAPPY SOFTWARE

präsentiert:

Commodore 64 — Super Action-Spiele auf Diskette

Mr. Robot
Action-Spiel
Ihre Aufgabe ist es hier, mit einem Roboter alle Power-Pills in einem Gewirr von Magneten, Trampolinen, Beamern, Bomben und lebendem Feuer aufzusammeln. Ganze 22 Levels beinhaltet dieses aufregende Spiel. Zum Glück hat unser Robi »Energizer«-Pillen, mit deren Hilfe er ungefähr durchkommt — doch die Anzahl ist begrenzt!
Best.-Nr. MD 221A (Sfr. 44,50) **DM 48,— ***

Lonely Rider
Action-Spiel
Indianer haben Ihren Kameraden gekidnappt. Sie, letzter der Kompanie, wagen sich direkt in die Höhle des Feindes, ins Indianercamp. In der Wüste, inmitten feindlichen Territoriums, müssen Sie und Ihr Pferd »Blacky« Ihre Mutprobe bestehen. Denn Indianer lauern überall. Überleben Sie durch reaktionsschnelle Ausweichmanöver! Ein Spiel für den Commodore 64 mit Diskettenlaufwerk.
Best.-Nr. MD 225A (Sfr. 44,50) **DM 48,— ***

Super Bunny
Action-Spiel
Reginald Rabbit, ein kleiner, schmächtiger Hase, macht eine merkwürdige Entdeckung: Zauberkarotten verleihen ihm Riesenkräfte. Jetzt ist Reggie Rabbit kein gewöhnlicher Hase mehr, sondern Super Bunny. Helfen Sie Reggie Rabbit in seinem Kampf für Freiheit und Gerechtigkeit! Erleben Sie seine atemberaubenden Einsätze!
Best.-Nr. MD 229A (Sfr. 44,50) **DM 48,— ***

Mastercode-Assembler
Mastercode ist ein vollständiges Programmpaket für die Entwicklung von Maschinenprogrammen. Neben dem eigentlichen Assembler sind noch verfügbar: ein Editor zur Eingabe von Quelltext · Ein Debugger, der Einzelschrittverarbeitung ermöglicht · ein Disassembler · Funktion zur Anzeige und zum Ändern des Speicherinhalts.
Best.-Nr. MK 110A (Sfr. 44,50) **DM 48,— ***
Best.-Nr. MD 110A (Sfr. 58,—) **DM 63,— ***

Aztec
Action-Adventure
Eines der neuesten und wirklich besten Action-Adventures unserer Zeit! Steuern Sie einen Abenteurer (mit 21 versch. Funktionen, Keyboard) in eine mörderische Pyramide. Darin befindet sich das sagenumwogene »IDOL«, das schon für viele den Tod bedeutete. Finden Sie sich zurecht, und ... vor allem finden Sie den Schatz.
Best.-Nr. MD 224A (Sfr. 44,50) **DM 48,— ***

Gladiator 2000
Action-Spiel
Bei »Gladiator« sind Sie der Fahrer eines heißen Ofens. Dieser hinterläßt, wenn er fährt, Reifenabdrücke, die für Ihren Gegner tödlich sind. Versuchen Sie, ihn in die Enge zu treiben, doch geraten Sie nicht selbst in eine solche Falle — sie hat tödliche Wirkung. Dabei kann der Gegner der Computer oder ein 2. Spieler sein. Der Spannung sind keine Grenzen gesetzt.
Best.-Nr. MD 227A (Sfr. 35,50) **DM 39,— ***

Cosmic Tunnels
Action-Spiel
Sie sind verantwortlich für den kleinen Planeten Sirref. Dieser durchlebt gerade eine akute Energieknappheit. Es fehlt nämlich der Grundbaustoff zum Betrieb der Kraftwerke. Ihre Aufgabe ist es nun, so viel wie möglich dieses Baustoffs von insgesamt 4 Asteroiden zu holen. Durchfliegen Sie 4 Zonen und bringen Sie das lebensnotwendige Material.
Best.-Nr. MD 222A (Sfr. 44,50) **DM 48,— ***

Catastrophes
Action-Spiel
Steigen Sie mit »Catastrophes« in das verrückteste Baugeschäft ein, das Sie je erlebt haben. Ihre Aufgabe ist es nämlich, mit Ihrem Helikopter ein Gebäude in die Welt zu setzen, das Hurricanes, Fluten und Erdbeben genauso widersteht wie dem Spieleifer eines 2. Spielers, bzw. des Computers! Sie haben nur 6 Tage Zeit, und es gibt viel zu tun!
Best.-Nr. MD 208A (Sfr. 44,50) **DM 48,— ***

Castle Nightmare
Action-Spiel
Steuern Sie Ihre Spielfigur durch ein unwegsames Schloß voller Hexen, Zauberer, Sensenmänner u.v.m. Sammeln Sie Lebenselexiere und Schlüssel, um in den nächsten Raum vorzudringen. Aber jeder beherbergt ein anderes Geheimnis! Seien Sie auf der Hut! Die Magie ist gegen Sie! Ein spannendes Spiel mit vielen Actionsscreens.
Best.-Nr. MD 226A (Sfr. 35,50) **DM 39,— ***

Photony
Action-Spiel
Dieses Spiel verbindet reine Action und raffinierte Strategie. Bekämpfen Sie ein feindliches Raumschiff mit einer beweglichen Laserkanone. Es wird aber zusätzlich von ebenfalls beweglichen Laserbasen beschützt. Der Clou dabei ist, daß das ganze Aktionsfeld übersät mit Spiegeln ist, die den tödlichen Laserstrahl reflektieren.
Best.-Nr. MD 228A (Sfr. 44,50) **DM 48,— ***

Markt&Technik
Verlag Aktiengesellschaft

Hans-Pinsel-Straße 2, 8013 Haar bei München, ☎ 089/4613-220
M&T-Vertriebs AG, Alpenstr. 14, CH-6300 Zug, ☎ 042-223155/56

In guten Buchhandlungen, Computershops und Fachabteilungen der Kaufhäuser.

* inkl. MwSt. Unverbindliche Preisempfehlung

MD = Diskette
MK = Kassette
A = Commodore 64

Ein besonderer Disassembler

Dieser Disassembler für den C 64 arbeitet ausschließlich mit Dezimalzahlen

Bei vielen Hobby-Programmierern besteht der einzige Kontakt zur Maschinensprache im gelegentlichen Eingeben von Basic-Ladern. Da tut man sich insbesondere mit den Hexadezimalzahlen schon mal etwas schwer.

Hier ist nun ein Disassembler, der sowohl Adressen als auch Befehls- und Datenbytes in dezimaler Form ausgibt.

Das Programm ist sehr einfach anzuwenden und erklärt sich im übrigen von selbst, so daß eine weitergehende Beschreibung hier überflüssig ist.

Bei der Adressenabfrage gibt es drei mögliche Eingaben:
1. Die direkte Anfangsadresse (dezimal)
2. »W« für weiter
3. »E« für Ende

Bei der folgenden Frage nach der Ausgabe auf Drucker kann man wählen zwischen:
1. »J« für Ja = Druckausgabe
2. »N» für Nein (Return reicht)

Das Programm läuft auf dem C 64, kann aber leicht auch für den VC 20 umgeschrieben werden (in Zeile 10 werden nur die Bildschirmfarben eingestellt).

Die verwendeten Steuerzeichen stellen nur Farben dar und können folglich frei gewählt werden.

(Stefan Heine / ev)

```
1 REM DEZIMAL-DISASSEMBLER                              <065>
2 REM                                                   <145>
3 REM STEFAN HEINE                                      <188>
4 REM WIESENREDDER 1                                    <069>
5 REM 2330 ECKERNFOERDE                                 <201>
6 REM                                                   <149>
7 REM                                                   <150>
10 POKE 53280,0:POKE 53281,0                            <047>
20 PRINT"[CLEAR,GREY 3,SPACE9]DISASSEMBLER
   [SPACE2]1.5"                                         <049>
30 PRINT"[SPACE12]BY[SPACE]ST.HEINE"                    <212>
40 PRINT:PRINT:PRINT                                    <103>
50 PRINT"[YELLOW]*[SPACE]ADRESSEN[SPACE]BITTE
   [SPACE]DEZIMAL[SPACE]"                               <170>
60 PRINT"[SPACE2]EINGEBEN!!!":PRINT                     <140>
70 PRINT"[GREEN]*[SPACE]ADRESSEN
   :[SPACE]W=WEITER[SPACE]E=ENDE":PRINT                 <207>
80 PRINT"*[SPACE]DRUCKER[SPACE]
   :[SPACE]'RETURN'=NEIN"                               <054>
90 PRINT:PRINT:PRINT                                    <153>
100 REM    COPYRIGHT * ST.HEINE *                       <062>
110 DIM M$(255),L%(255),F%(255),F1$(12),F2$(12)
                        <087>
120 FOR I=0 TO 255:READ M$(I):NEXT
   :FOR I=0 TO 255:READ L%(I):NEXT
   :FOR I=0 TO 255:READ F%(I)                           <134>
130 NEXT:FOR I=0 TO 12:READ F1$(I),F2$(I):NEXT
   :PC=0                                                <180>
140 IF IC=1 THEN GOSUB 360:IC=0                         <251>
150 INPUT"[LIG.BLUE]#[SPACE]ANFANGSADRESSE";A$
                        <193>
160 INPUT"[GREY 3]#[SPACE]AUSGABE[SPACE]AUF
   [SPACE]DRUCKER";GC$                                  <244>
170 HC$=LEFT$(GC$,1)                                    <047>
180 IF HC$="J"THEN IC=1                                 <068>
190 J=LEN(A$):IF J>5 THEN 140                           <100>
200 IF A$="W"THEN 230                                   <065>
210 IF A$="E"THEN 900                                   <061>
220 PC=VAL(A$)                                          <156>
230 FOR I=1 TO 24:DZ=PC:PRINT"[GREEN]";
   :P1=PEEK(PC)                                         <061>
240 ON L%(P1)GOSUB 260,270,330                          <112>
250 NEXT:GOTO 140                                       <212>
260 PRINT TAB(1)"[RED]";PC;:PRINT TAB(9)"[GREEN]
   ";P1;:PRINT"[BLUE]";TAB(24);M$(P1):PC=PC+1
   :RETURN                                              <011>
270 PRINT TAB(1)"[RED]";PC;:PRINT TAB(9)"[GREEN]
   ";P1;:P2=PEEK(PC+1):PRINT TAB(13)P2;                 <238>
280 PRINT"[BLUE]";TAB(24)M$(P1);"[SPACE,
   LIG.BLUE]";                                          <025>
290 IF F%(P1)=9 THEN P2=PEEK(PC+1)
   :IF P2<128 THEN P2=PC+P2+2:GOTO 310                  <148>
300 IF F%(P1)=9 THEN P2=PC-(255-P2)+1                   <238>
310 PRINT F1$(F%(P1));P2;:DZ=P2:GOTO 320                <085>
320 PRINT F2$(F%(P1)):PC=PC+2:RETURN                    <186>
330 PRINT TAB(1)"[RED]";PC;:PRINT TAB(9)"[GREEN]
   ";P1;:P2=PEEK(PC+1):PRINT TAB(14)P2;                 <043>
340 P3=PEEK(PC+2):PRINT TAB(19)P3;
   :PRINT"[BLUE]";TAB(24);M$(P1)"[LIG.BLUE]";
   :P4=(P3*256)+P2                                      <198>
350 PRINT P4;F1$(F%(P1));:DZ=P3*256+P2
   :PRINT F2$(F%(P1)):PC=PC+3:RETURN                    <081>
360 OPEN 4,4:PRINT#4,CHR$(15)                           <076>
370 VB=1024:VC=55296                                    <020>
380 FOR I=VB TO VB+999 STEP 40:P$=""                    <211>
390 FOR J=0 TO 39:X=PEEK(I+J)                           <216>
400 IF X<32 THEN X=X+64:GOTO 440                        <004>
410 IF X<64 THEN 440                                    <218>
420 IF X<96 THEN X=X+32:GOTO 440                        <029>
430 IF X<128 THEN X=X+64:GOTO 440                       <088>
440 P$=P$+CHR$(X)                                       <109>
450 NEXT:PRINT#4,P$:NEXT:CLOSE 4:RETURN                 <125>
460 REM*** DATAS BEFEHLSLISTE ***                       <063>
470 DATA BRK,ORA,???,???,???,ORA,ASL,???,PHP,
   ORA,ASL,???,???,ORA,ASL,???                          <008>
480 DATA BPL,ORA,???,???,???,ORA,ASL,???,CLC,
   ORA,???,???,???,ORA,ASL,???                          <216>
490 DATA JSR,AND,???,???,BIT,AND,ROL,???,PLP,
   AND,ROL,???,BIT,AND,ROL,???                          <095>
500 DATA BMI,AND,???,???,???,AND,ROL,???,SEC,
   AND,???,???,???,AND,ROL,???                          <205>
510 DATA RTI,EOR,???,???,???,EOR,LSR,???,PHA,
   EOR,LSR,???,JMP,EOR,LSR,???                          <158>
520 DATA BVC,EOR,???,???,???,EOR,LSR,???,CLI,
   EOR,???,???,???,EOR,LSR,???                          <054>
530 DATA RTS,ADC,???,???,???,ADC,ROR,???,PLA,
   ADC,ROR,???,JMP,ADC,ROR,???                          <079>
540 DATA BVS,ADC,???,???,???,ADC,ROR,???,SEI,
   ADC,???,???,???,ADC,ROR,???                          <239>
550 DATA???,STA,???,???,STY,STA,STX,???,DEY,???,
   TXA,???,STY,STA,STX,???                              <239>
560 DATA BCC,STA,???,???,STY,STA,STX,???,TYA,
   STA,TXS,???,???,STA,???,???                          <200>
570 DATA LDY,LDA,LDX,???,LDY,LDA,LDX,???,TAY,
   LDA,TAX,???,LDY,LDA,LDX,???                          <217>
580 DATA BCS,LDA,???,???,LDY,LDA,LDX,???,CLV,
   LDA,TSX,???,LDY,LDA,LDX,???                          <176>
590 DATA CPY,CMP,???,???,CPY,CMP,DEC,???,INY,
   CMP,DEX,???,CPY,CMP,DEC,???                          <197>
600 DATA BNE,CMP,???,???,???,CMP,DEC,???,CLD,
   CMP,???,???,???,CMP,DEC,???                          <025>
610 DATA CPX,SBC,???,???,CPX,SBC,INC,???,INX,
   SBC,NOP,???,CPX,SBC,INC,???                          <221>
620 DATA BEQ,SBC,???,???,???,SBC,INC,???,SED,
   SBC,???,???,???,SBC,INC,???                          <053>
630 REM*** DATAS 1,2 ODER 3 BIT ***                     <059>
640 DATA 1,2,1,1,1,2,2,1,1,2,1,1,1,3,3,1,2,2,1,
   1,1,2,2,1,1,3,1,1,1,3,3,1                            <139>
650 DATA 3,2,1,1,2,2,2,1,1,2,1,1,3,3,3,1,2,2,1,
   1,1,2,2,1,1,3,1,1,1,3,3,1                            <154>
660 DATA 1,2,1,1,1,2,2,1,1,2,1,1,3,3,3,1,2,2,1,
   1,1,2,2,1,1,3,1,1,1,3,3,1                            <161>
670 DATA 1,2,1,1,1,2,2,1,1,2,1,1,3,3,3,1,2,2,1,
```

```
   1,1,2,2,1,1,3,1,1,1,3,3,1                 <171>
680 DATA 1,2,1,1,2,2,2,1,1,1,1,1,3,3,3,1,2,2,1,
   1,2,2,2,1,1,3,1,1,1,3,1,1                 <180>
690 DATA 2,2,2,1,2,2,2,1,1,2,1,1,3,3,3,1,2,2,1,
   1,2,2,2,1,1,3,1,1,3,3,3,1                 <197>
700 DATA 2,2,1,1,2,2,2,1,1,2,1,1,3,3,3,1,2,2,1,
   1,1,2,2,1,1,3,1,1,1,3,3,1                 <203>
710 DATA 2,2,1,1,2,2,2,1,1,2,1,1,3,3,3,1,2,2,1,
   1,1,2,2,1,1,3,1,1,1,3,3,1                 <213>
720 REM*** DATAS BEFEHLART (ZEICHEN)***      <174>
730 DATA 0,10,0,0,0,4,4,0,0,2,1,0,0,3,3,0  <053>
740 DATA 9,11,0,0,0,5,5,0,0,8,0,0,0,7,7,0  <088>
750 DATA 3,10,0,0,4,4,4,0,0,2,1,0,3,3,3,0  <083>
760 DATA 9,11,0,0,0,5,5,0,0,8,0,0,0,7,7,0  <108>
770 DATA 0,10,0,0,0,4,4,0,0,2,1,0,3,3,3,0  <097>
780 DATA 9,11,0,0,0,5,5,0,0,8,0,0,0,7,7,0  <129>
790 DATA 0,10,0,0,0,4,4,0,0,2,1,0,12,3,3,0 <165>
800 DATA 9,11,0,0,0,5,5,0,0,8,0,0,0,7,7,0  <117>
810 DATA 0,10,0,0,4,4,4,0,0,0,0,0,3,3,3,0  <138>
820 DATA 9,11,0,0,5,5,6,0,0,8,0,0,0,7,0,0  <168>
830 DATA 2,10,2,0,4,4,4,0,0,2,0,0,3,3,3,0  <164>
840 DATA 9,11,0,0,5,5,6,0,0,8,0,0,7,7,8,0  <203>
850 DATA 2,10,0,0,4,4,4,0,0,2,0,0,3,3,3,0  <182>
860 DATA 9,11,0,0,0,5,5,0,0,8,0,0,0,7,7,0  <209>
870 DATA 2,10,0,0,4,4,4,0,0,2,0,0,3,3,3,0  <202>
880 DATA 9,11,0,0,0,5,5,0,0,8,0,0,0,7,7,0  <229>
890 DATA,,,,#,,,,,,,,",X",,",Y",,",X",,",Y",,,(,
   ",X)",(,"),Y",(,),                        <065>
900 END                                      <007>
```

Mouse 64

Das Programm Mouse 64 stellt eine Interrupt-Erweiterung des C64 dar und verträgt sich somit nicht ohne eine Anpassung mit anderen Interrupt-Erweiterungen. Das Programm erzeugt nach Aufruf eine »Maus« auf dem Bildschirm, die sich mit einem Joystick (Port II) steuern läßt. Die Abfrage erfolgt durch eine in den Interrupt eingebundene Routine, so daß die Tastatur weiterhin benutzbar bleibt.

Mouse 64 ist vollständig in 6502/10-Assembler geschrieben und benötigt keinen Basic-Speicher, da Mouse 64 im freien RAM-Bereich von 49152 ($c000) bis 53272 ($cfff) liegt. Die Sprite-Daten liegen in Block 11 (dez. 704-767). Um die Maus darzustellen, wird Sprite 7 verwendet. Das Programm liefert folgende Daten:

a) Den Bildschirmcode des Zeichens unter der Maus
b) Die Bildschirm-X-Koordinate
c) Die Bildschirm-Y-Koordinate
d) Joystick # 2-Zustand
e) Feuerknopf # 2-Zustand

Das Programm Mouse 64 ist als Eingabe- und Steuerhilfe für Anwendungsprogramme gedacht (siehe Apple-Macintosh-Software und andere), kann aber auch durch zum Beispiel Änderung des Sprites in Spiele eingebunden werden, da bis auf die Vergrößerung alle Spriteparameter veränderlich bleiben.

Zu Listing 1

Das Listing 1 zeigt den Basic-Lader, der in zwei Schleifen die Daten für den Sprite und das Assemblerprogramm aus den DATA-Zeilen liest und in die entsprechenden Speicherstellen schreibt.

Bei einem Fehler der Daten bricht der Computer das Programm mit einer entsprechenden Meldung ab.

Nach Ablauf der beiden Schleifen wird das Assemblerprogramm direkt mit »SYS 49152« gestartet.

Zu Listing 2

Das zweite Listing zeigt die in Assembler geschriebene Initialisierungs-Routine. Diese Routine verändert die Interrupt-Zeiger so, daß sie auf die Maus-Routine zeigen. Ferner verändert sie die zu Sprite 7 gehörenden Parameter entsprechend.

Die Initialisierungsroutine gibt den Text:
PUT JOYSTICK IN PORT 2
AND PRESS ANY KEY
aus. Wenn man diesem Folge geleistet hat erscheint eine »Maus« in der linken oberen Ecke und der Computer arbeitet normal im Programm beziehungsweise im Direktmodus weiter. Die Maus läßt sich — wie gesagt — mit dem Joystick # 2 steuern.

Zu Listing 3

Listing 3 zeigt die eigentliche Interrupt-Erweiterung. Diese beginnt mit dem Auslesen des Joystickzustandes. Dazu muß bekanntlich die Tastatur abgeschaltet werden. Das Programm schaltet also die Tastatur ab, liest das Register für den 2. Joystick aus und schaltet die Tastatur wieder ein. So bleibt die Tastatur weiterhin benutzbar.

Daraufhin wird anhand der Joystickstellung der Sprite 7, also die Maus, bewegt. Ist dies erfolgt, werden aus den Spritekoordinaten die Bildschirmkoordinaten errechnet. Sind diese bekannt, so wird daraus das entsprechende Byte unter der Maus errechnet und ausgelesen.

Diese Werte sind folgenden Registern zu entnehmen:

1. Der Zeichencode des unter der Maus befindlichen Zeichens: 49659 ($c1fb)
2. Die Bildschirm-X-Koordinate : 49660 ($c1fc)
3. Die Bildschirm-Y-Koordinate : 49661 ($c1fd)
4. Der Feuerknopf # 2 : 49662 ($c1fe) (=1: gedrückt / = 0 nicht gedrückt)
5. Der Zustand des Joysticks : 49663 ($c1ff)

Zu Listing 4

Listing 4 zeigt den Hex-Dump des Assembler-Programms. Abschließend wäre noch zu sagen, daß es keinen Sinn hat, zu versuchen, das Programm anderer Computer, die nicht wie der C 64 in der Lage sind, frei bewegliche Sprites zu erzeugen, da ein solcher Sprite nötig ist.

(Peter Dreuw / rg)

Listing 1. Der Basic-Lader der «Mouse 64«

```
0 REM  ***********************              <043>
1 REM  *       MOUSE 64        *              <215>
2 REM  *                       *              <229>
3 REM  *       1984  BY        *              <087>
4 REM  *                       *              <231>
5 REM  *     PETER DREUW       *              <239>
6 REM  * MENNRATHSCHMIDT 27 *                 <187>
7 REM  *                       *              <234>
8 REM  * MOENCHENGLADBACH 5 *                 <147>
9 REM  *                       *              <236>
10 REM ***********************               <053>
20 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0      <043>
30 DATA 48,0,0,120,0,0,252,0,0,252,0,1,254,0,1,
   254                                        <126>
40 DATA 0,0,252,0,0,48,0,0,16,0,0,16,0,0,32,0
                    <135>
50 DATA 0,64,0,0,0,0,0,0,0,0,0,0,0,0,0     <131>
60 FOR I=704 TO 767:READ Q:POKE I,Q:S=S+Q:NEXT
                    <141>
70 IF S<>1610 THEN PRINT"MOUSE-DATA'S[SPACE]
   FALSCH":END                                <137>
80 PRINT"MOUSE-DATA[SPACE]OK":S=0            <006>
85 REM PRG-DATA                               <020>
90 FOR I=49152 TO 49555:READ Q:POKE I,Q:S=S+Q
   :NEXT                                      <125>
91 DATA 160,231,185,117,191,32,210,255,200,208,
   247,160,238,185,134,191,32                 <100>
92 DATA 210,255,200,208,247,165,198,201,0,240,
   250,169,0,133,198,120,169,137              <200>
93 DATA 141,20,3,169,192,141,21,3,169,17,141,14,
   208,169,45,141,15,208,173                  <098>
```

```
94 DATA 21,208,9,128,141,21,208,173,27,208,41,
   127,141,27,208,173,28,208,41                <151>
95 DATA 127,141,28,208,169,0,141,46,208,169,11,
   141,255,7,88,96,0,0,255,80                  <114>
96 DATA 85,84,32,74,79,89,83,84,73,67,75,32,73,
   78,32,80,79,82,84,32,50,32                  <163>
97 DATA 33,13,65,78,68,32,80,82,69,83,83,32,65,
   78,89,32,75,69,89,0,0,0,162                 <196>
98 DATA 224,142,2,220,173,0,220,162,255,142,2,
   220,141,255,193,41,1,208,10                 <067>
99 DATA 173,15,208,201,45,144,3,206,15,208,173,
   255,193,41,2,208,10,169,236                 <154>
100 DATA 205,15,208,144,3,238,15,208,173,255,
   193,41,4,208,41,173,16,208,201              <154>
101 DATA 128,144,24,173,14,208,240,5,206,14,208,
   208,24,173,16,208,41,127,141                <199>
102 DATA 16,208,206,14,208,76,232,192,173,14,
   208,201,18,144,3,206,14,208,173             <205>
103 DATA 255,193,41,8,208,33,173,16,208,201,128,
   144,13,173,14,208,201,72,240                <203>
104 DATA 19,238,14,208,76,16,193,238,14,208,208,
   8,173,16,208,9,128,141,16                   <085>
105 DATA 208,162,0,173,255,193,41,16,208,1,232,
   142,254,193,173,23,208,41,127               <209>
106 DATA 141,23,208,173,29,208,41,127,141,29,
   208,173,15,208,160,255,56,200               <115>
107 DATA 233,8,176,251,152,233,4,168,140,253,
   193,173,16,208,201,128,144,20               <116>
108 DATA 173,14,208,162,255,232,233,8,176,251,
   138,105,30,170,142,252,193,76               <174>
109 DATA 108,193,56,173,14,208,162,255,232,233,
   8,176,251,138,233,1,170,142                 <125>
110 DATA 252,193,169,4,133,252,234,134,251,192,
   0,240,16,24,169,40,101,251                  <061>
111 DATA 133,251,165,252,105,0,133,252,136,208,
   240,177,251,141,251,193,165                 <116>
112 DATA 252,76,49,234,0,0,0                   <151>
120 IF S<>52007 THEN PRINT"PRG-DATA'S[SPACE]
   FALSCH":END                                 <081>
130 PRINT"PRG-DATA[SPACE]OK"                   <041>
140 PRINT"PRESS[SPACE]ANY[SPACE]KEY[SPACE]TO
   [SPACE]START                                <214>
150 POKE 198,0:WAIT 198,1                      <246>
160 SYS 49152                                  <067>
```

Listing 2. Die in Assembler geschriebene Initialisierungsroutine von »Mouse 64«

```
LISTING 2

.
.,  c000  a0 e7     ldy #$e7
.,  c002  b9 75 bf  lda $bf75,y
.,  c005  20 d2 ff  jsr $ffd2
.,  c008  c8        iny
.,  c009  d0 f7     bne $c002
.,  c00b  a0 ee     ldy #$ee
.,  c00d  b9 86 bf  lda $bf86,y
.,  c010  20 d2 ff  jsr $ffd2
.,  c013  c8        iny
.,  c014  d0 f7     bne $c00d
.,  c016  a5 c6     lda $c6
.,  c018  c9 00     cmp #$00
.,  c01a  f0 fa     beq $c016
.,  c01c  a9 00     lda #$00
.,  c01e  85 c6     sta $c6
.,  c020  78        sei
.,  c021  a9 89     lda #$89
.,  c023  8d 14 03  sta $0314
.,  c026  a9 c0     lda #$c0
.,  c028  8d 15 03  sta $0315
.,  c02b  a9 11     lda #$11
.,  c02d  8d 0e d0  sta $d00e
.,  c030  a9 2d     lda #$2d
.,  c032  8d 0f d0  sta $d00f
.,  c035  ad 15 d0  lda $d015
.,  c038  09 80     ora #$80
.,  c03a  8d 15 d0  sta $d015
.,  c03d  ad 1b d0  lda $d01b
.,  c040  29 7f     and #$7f
.,  c042  8d 1b d0  sta $d01b
.,  c045  ad 1c d0  lda $d01c
.,  c048  29 7f     and #$7f
.,  c04a  8d 1c d0  sta $d01c
.,  c04d  a9 00     lda #$00
.,  c04f  8d 2e d0  sta $d02e
.,  c052  a9 0b     lda #$0b
.,  c054  8d ff 07  sta $07ff
.,  c057  58        cli
.,  c058  60        rts
.
```

Listing 3. Die eigentliche Interrupt-Erweiterung von »Mouse 64«

```
LISTING 3

.,  c089  a2 e0     ldx #$e0
.,  c08b  8e 02 dc  stx $dc02
.,  c08e  ad 00 dc  lda $dc00
.,  c091  a2 ff     ldx #$ff
.,  c093  8e 02 dc  stx $dc02
.,  c096  8d ff c1  sta $c1ff
.,  c099  29 01     and #$01
.,  c09b  d0 0a     bne $c0a7
.,  c09d  ad 0f d0  lda $d00f
.,  c0a0  c9 2d     cmp #$2d
.,  c0a2  90 03     bcc $c0a7
.,  c0a4  ce 0f d0  dec $d00f
.,  c0a7  ad ff c1  lda $c1ff
.,  c0aa  29 02     and #$02
.,  c0ac  d0 0a     bne $c0b8
.,  c0ae  a9 ec     lda #$ec
.,  c0b0  cd 0f d0  cmp $d00f
.,  c0b3  90 03     bcc $c0b8
.,  c0b5  ee 0f d0  inc $d00f
.,  c0b8  ad ff c1  lda $c1ff
.,  c0bb  29 04     and #$04
.,  c0bd  d0 29     bne $c0e8
.,  c0bf  ad 10 d0  lda $d010
.,  c0c2  c9 80     cmp #$80
.,  c0c4  90 18     bcc $c0de
.,  c0c6  ad 0e d0  lda $d00e
.,  c0c9  f0 05     beq $c0d0
.,  c0cb  ce 0e d0  dec $d00e
.,  c0ce  d0 18     bne $c0e8
.,  c0d0  ad 10 d0  lda $d010
.,  c0d3  29 7f     and #$7f
.,  c0d5  8d 10 d0  sta $d010
.,  c0d8  ce 0e d0  dec $d00e
.,  c0db  4c e8 c0  jmp $c0e8
.,  c0de  ad 0e d0  lda $d00e
.,  c0e1  c9 12     cmp #$12
```

```
.,    c0e3   90 03      bcc $c0e8
.,    c0e5   ce 0e d0   dec $d00e
.,    c0e8   ad ff c1   lda $c1ff
.,    c0eb   29 08      and #$08
.,    c0ed   d0 21      bne $c110
.,    c0ef   ad 10 d0   lda $d010
.,    c0f2   c9 80      cmp #$80
.,    c0f4   90 0d      bcc $c103
.,    c0f6   ad 0e d0   lda $d00e
.,    c0f9   c9 48      cmp #$48
.,    c0fb   f0 13      beq $c110
.,    c0fd   ee 0e d0   inc $d00e
.,    c100   4c 10 c1   jmp $c110
.,    c103   ee 0e d0   inc $d00e
.,    c106   d0 08      bne $c110
.,    c108   ad 10 d0   lda $d010
.,    c10b   09 80      ora #$80
.,    c10d   8d 10 d0   sta $d010
.,    c110   a2 00      ldx #$00
.,    c112   ad ff c1   lda $c1ff
.,    c115   29 10      and #$10
.,    c117   d0 01      bne $c11a
.,    c119   e8         inx
.,    c11a   8e fe c1   stx $c1fe
.,    c11d   ad 17 d0   lda $d017
.,    c120   29 7f      and #$7f
.,    c122   8d 17 d0   sta $d017
.,    c12a   8d 1d d0   sta $d01d
.,    c12d   ad 0f d0   lda $d00f
.,    c130   a0 ff      ldy #$ff
.,    c132   38         sec
.,    c133   c8         iny
.,    c134   e9 08      sbc #$08
.,    c136   b0 fb      bcs $c133
.,    c138   98         tya
.,    c139   e9 04      sbc #$04
.,    c13b   a8         tay
.,    c13c   8c fd c1   sty $c1fd
.,    c13f   ad 10 d0   lda $d010
.,    c142   c9 80      cmp #$80
.,    c144   90 14      bcc $c15a
.,    c146   ad 0e d0   lda $d00e
.,    c149   a2 ff      ldx #$ff
.,    c14b   e8         inx
.,    c14c   e9 08      sbc #$08
.,    c14e   b0 fb      bcs $c14b
.,    c150   8a         txa
.,    c151   69 1e      adc #$1e
.,    c153   aa         tax
.,    c154   8e fc c1   stx $c1fc
.,    c157   4c 6c c1   jmp $c16c
.,    c15a   38         sec
.,    c15b   ad 0e d0   lda $d00e
.,    c15e   a2 ff      ldx #$ff
.,    c160   e8         inx
.,    c161   e9 08      sbc #$08
.,    c163   b0 fb      bcs $c160
.,    c165   8a         txa
.,    c166   e9 01      sbc #$01
.,    c168   aa         tax
.,    c169   8e fc c1   stx $c1fc
.,    c16c   a9 04      lda #$04
.,    c16e   85 fc      sta $fc
.,    c170   ea         nop
.,    c171   86 fb      stx $fb
.,    c173   c0 00      cpy #$00
.,    c175   f0 10      beq $c187
.,    c177   18         clc
.,    c178   a9 28      lda #$28
.,    c17a   65 fb      adc $fb
.,    c17c   85 fb      sta $fb
.,    c17e   a5 fc      lda $fc
.,    c180   69 00      adc #$00
.,    c182   85 fc      sta $fc
.,    c184   88         dey
.,    c185   d0 f0      bne $c177
.,    c187   b1 fb      lda ($fb),y
.,    c189   8d fb c1   sta $c1fb
.,    c18c   a5 fc      lda $fc
.,    c18e   4c 31 ea   jmp $ea31
.,    c191   00         brk
.
```

Listing 4. Der Hex-Dump des Assemblerlisting von »Mouse 64«

LISTING 4

```
.
.:  c000 a0 e7 b9 75 bf 20 d2 ff c8 d0 f7 a0 ee b9 86 bf
.:  c010 20 d2 ff c8 d0 f7 a5 c6 c9 00 f0 fa a9 00 85 c6
.:  c020 78 a9 89 8d 14 03 a9 c0 8d 15 03 a9 11 8d 0e d0
.:  c030 a9 2d 8d 0f d0 ad 15 d0 09 80 8d 15 d0 ad 1b d0
.:  c040 29 7f 8d 1b d0 ad 1c d0 29 7f 8d 1c d0 a9 00 8d
.:  c050 2e d0 a9 0b 8d ff 07 58 60 00 00 ff 50 55 54 20
.:  c060 4a 4f 59 53 54 49 43 4b 20 49 4e 20 50 4f 52 54
.:  c070 20 32 20 21 0d 41 4e 44 20 50 52 45 53 53 20 41
.:  c080 4e 59 20 4b 45 59 00 00 00 a2 e0 8e 02 dc ad 00
.:  c090 dc a2 ff 8e 02 dc 8d ff c1 29 01 d0 0a ad 0f d0
.:  c0a0 c9 2d 90 03 ce 0f d0 ad ff c1 29 02 d0 0a a9 ec
.:  c0b0 cd 0f d0 90 03 ee 0f d0 ad ff c1 29 04 d0 29 ad
.:  c0c0 10 d0 c9 80 90 18 ad 0e d0 f0 05 ce 0e d0 d0 18
.:  c0d0 ad 10 d0 29 7f 8d 10 d0 ce 0e d0 4c e8 c0 ad 0e
.:  c0e0 d0 c9 12 90 03 ce 0e d0 ad ff c1 29 08 d0 21 ad
.:  c0f0 10 d0 c9 80 90 0d ad 0e d0 c9 48 f0 13 ee 0e d0
.:  c100 4c 10 c1 ee 0e d0 d0 08 ad 10 d0 09 80 8d 10 d0
.:  c110 a2 00 ad ff c1 29 10 d0 01 e8 8e fe c1 ad 17 d0
.:  c120 29 7f 8d 17 d0 ad 1d d0 29 7f 8d 1d d0 ad 0f d0
.:  c130 a0 ff 38 c8 e9 08 b0 fb 98 e9 04 a8 8c fd c1 ad
.:  c140 10 d0 c9 80 90 14 ad 0e d0 a2 ff e8 e9 08 b0 fb
.:  c150 8a 69 1e aa 8e fc c1 4c 6c c1 38 ad 0e d0 a2 ff
.:  c160 e8 e9 08 b0 fb 8a e9 01 aa 8e fc c1 a9 04 85 fc
.:  c170 ea 86 fb c0 00 f0 10 18 a9 28 65 fb 85 fb a5 fc
.:  c180 69 00 85 fc 88 d0 f0 b1 fb 8d fb c1 a5 fc 4c 31
.:  c190 ea 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4a
.
■
```

17 Super-Utilities für den C 64

Wollen Sie Ihre Basic-Programme schneller und kürzer machen? Wollen Sie Betriebssystem oder Zeichensatz Ihres C 64 ändern? Dieses Maschinenprogramm löst Ihre Probleme.

Das Programm entstand durch Zusammenfügen der in der praktischen Arbeit am häufigsten gebrauchten Maschinensprache-Routinen zu einem einzigen großen Utility-Paket.

Die beiden Bereiche Optimierung von Basic-Programmen und Zeichen-Definition stehen im Mittelpunkt. Nach ordnungsgemäßem Laden kann man ein Menü mit »sys 49152« aufrufen (Bild 1), in dem alle Unterprogramme mit Namen und SYS-Adresse genannt werden.

Nun zu den Programmen selbst:

Space-Killer (sys 49155): Hinter diesem martialisch anmutenden Aufruf verbirgt sich nichts anderes als eine Maschinen-Routine, welche überflüssige Spaces aus einem Basic-Programm entfernt.

Nach dem »sys 49155«-Aufruf erscheint in der oberen Bildschirmecke ein farbiges Symbol als Versicherung, daß alles in Ordnung ist, denn die Routine kann durchaus bis zu 1/4 Stunde laufen, wenn viele Spaces zu entfernen sind und das Programm entsprechend lang ist.

Space-Expander (sys 49158): Es erscheint auf den ersten Blick befremdend, etwas einzufügen, was man eben erst entfernt hatte. Doch die Aufgabe liegt auf der Hand: zwischen jeden Basic-Befehl ein Space einzufügen, um das Druckbild sauber und ansprechend zu gestalten.

REM-Killer (sys49161): erklärt sich selbst. Erwähnenswert ist die Geschwindigkeit, mit der die Routine läuft. 20 kByte-Programme werden in etwa 10-30 Sekunden »gesäubert«.

Zeilen löschen (sys 49164): Hiermit werden ganze Zeilenbereiche gelöscht. Beispiel: »sys 49164,1000,1999« entfernt alle Basic-Zeilen mit den Nummern von 1000 bis 1999 einschließlich. Mit Druck auf die RETURN-Taste ist der Befehl schon ausgeführt.

Text suchen (sys 49167,"Suchtext"): Sucht eine Zeichenkette »Suchtext« im Basic-Programm und gibt die gefundenen Zeilennummern aus.

Eine Besonderheit ist das Jokerzeichen »?«: Hier wird jedes Zeichen akzeptiert, es darf allerdings nicht an erster Stelle im Suchtext stehen.

Colon-Maker (sys 49170): Ersetzt jedes Space am Anfang einer Basic-Zeile durch einen Doppelpunkt »:«.

Space-Maker (sys 49173): Ersetzt jeden Doppelpunkt am Anfang einer Basic-Zeile durch ein Space » «. Beide Routinen dienen als Strukturhilfen. Da alle der Zeilennummer unmittelbar folgenden Spaces überlesen werden, muß man erst einen »neutralen« Doppelpunkt setzen, diesen dann später durch ein Space ersetzen.

Package (sys 49179): Hier ist die Notlösung für jeden, der einfach mehr Platz braucht und das Basic-Programm garantiert nicht mehr korrigieren muß. »Package« schmilzt nämlich benachbarte Zeilen zusammen, so daß bei jeder Verschmelzung vier Bytes freiwerden (Ein Null-Byte, Zwei Bytes der Koppeladresse plus Zwei Bytes der Zeilennummer, abzüglich ein Byte für den nun erforderlichen Doppelpunkt als Trennelement).
Das braucht seine Zeit, weil einige Aspekte kontrolliert werden müssen, zum Beispiel ob die zu verschmelzende Zeile angesprungen wird, dann muß sie mit ihrer Zeilennummer präsent bleiben, sonst gibt's Ärger mit dem Interpreter, der sich dann mit »undef'd statement« beschwert. Für die Verschmelzungstiefe gibt es leider eine unnatürliche Grenze: Mehr als 255 Bytes darf eine Zeile nicht lang sein, sonst kann der Interpreter die Zeilen nicht mehr binden und findet hinter einer solchen »Megazeile« keine Zeilennummer mehr.
Um Basic-Programme auf ein Minimum zu komprimieren, empfiehlt sich folgender Weg:
— zuerst »sys 49161« eingeben. Danach sind alle REMarks weg, und die folgenden Routinen haben weniger Prüfarbeit.
— dann »sys 49155« eingeben. Hier werden alle Zeilen gestaucht.
— zuletzt »sys 49179« eingeben, danach ist Ihr PRG redundanzfrei.

Für spätere Änderungsarbeiten wird man sich natürlich ein »augenfreundliches« Original verwahren, am besten benutzt man von allen Basic-Programmen zwei Versionen: Eine originale zur Dokumentation, Korrektur, Ausdruck und eine komprimierte mit minimalen Platzansprüchen und schneller Ausführung.

CPU-Register ein (sys 49182): Zeigt fortwährend die Register des Mikroprozessors in den beiden oberen Bildschirmzeilen an.
Außerdem werden der IRQ-, NMI- und der BRK-Vektor sowie der Programmcounter angezeigt. Im Status-Register bedeutet »*« ein gesetztes Bit, ».« ein gelöschtes.

CPU-Register aus (sys 49185): Schaltet die Anzeige wieder ab.

Basic-Ende setzen (sys 49188): setzt das Basic-Ende genau hinter das im Speicher befindliche Programm. Normalerweise erledigt das der Interpreter; wenn aber absolut geladen wurde, zum Beispiel in den $c-Block hinein, dann steht der Programm-Ende-Zeiger ebenfalls im $c-Block, und Eingabeversuche werden mit »out of memory«-Fehlermeldungen abgewehrt.

Dump (sys 49191): Gibt alle definierten Variablen unter Angabe des Namens und des aktuellen Inhalts aus.
Mit der Shift-Taste kann die Ausgabe angehalten, mit der Commodore-Taste vorzeitig abgebrochen werden.

ROM kopieren (sys 49194): Kopiert Interpreter- und Kernal-ROM in das darunter liegende RAM. Es gibt zwar viele veröffentlichte Programme zu diesem Thema, doch macht die Geschwindigkeit dieser Stilblüten auf Dauer aggressiv.

Zeichensatz RAM Die letzten vier Routinen befassen sich mit dem Zeichensatz:
(sys 49197): Verlegt den Zeichensatz nach $7000 und den Bildschirm nach $6c00. Die Absicht ist klar: im RAM kann der Zeichensatz leicht verändert werden.

Zeichensatz ändern (sys 49200): Setzt voraus, daß vorher durch »sys 49197« der Zeichensatz verlegt wurde, sonst funktioniert's nicht. Es wird eine Zeichen-Änderungsroutine akti-

viert, die einfach zu bedienen ist. Man fährt einfach mit dem Cursor über das zu ändernde Zeichen, drückt RETURN, und schon guckt man wie durch eine Lupe in die Punkte-Matrix des Zeichens hinein. Auch hier kann man sich wieder frei mit dem Cursor bewegen, natürlich nur innerhalb des Zeichens. Will man einen Punkt löschen, drückt man die Space-Taste, zum Setzen gibt man ein Sternchen »*« ein. RETURN führt wieder in das Menü zurück.

Zeichensatz speichern (sys 49203, "filename", gerätenummer)
Speichert den aktuellen Zeichensatz unter Angabe des Filenamens (der wie bei »SAVE« dem SYS-Befehl in Anführungsstrichen folgen muß), eines Kommas und der Gerätenummer entweder auf der Floppy oder dem Kassettenrecorder ab. Damit nicht jedesmal wieder bei Null begonnen werden muß, kann man einen Zeichensatz durch den Basic-Befehl:
»load "filename", gerätenummer,1«
absolut, das heißt von $7000 bis $7fff laden.
Die verbogenen Basic-Zeiger werden danach mit »sys 49188« wieder gerade gerichtet.
Zeichensatz ROM (sys 49206): Schaltet den Zeichensatz wieder ins ROM $d000 zurück. Der Bildschirm wird wieder nach $400 verlegt.

Grundsätzliches zur Eingabe

Maschinenprogramme sind extrem empfindlich, was Fehler anbetrifft. Einfache Prüfsummen decken sich ausgleichende Fehler nicht auf. Wird beispielsweise eine »1« und eine »2« vertauscht, so stimmt die Prüfsumme zwar, das Programm ist aber wahrscheinlich nicht lauffähig. Aus diesem Grund habe ich die Prüfsummen gespalten, »pg« für gerade Bytes und »pu« für ungerade. Zwar sind auch jetzt noch Fehlermaskierungen möglich, aber extrem unwahrscheinlich.

Noch eine Unart erregt manches Ärgernis:
Viele Autoren veröffentlichen Basic-Lader, und geben dann nach etwa 4000 harten (das heißt unformatierten) Dezimal-Bytes eine (also eine einzige) Prüfsumme an, so ziemlich weit in den Hunderttausendern, die auf Anhieb nur selten übereinstimmt. Es wollen dann 4000 Bytes überprüft werden! Ich halte es für besser, nach einem kleineren Abschnitt, etwa einer »Page« (256 Bytes) eine Zwischenprüfung zu fahren und bei Unstimmigkeiten sofort mit einem Hinweis zu unterbrechen.

In meinem Programm werden den Datenbytes die Angabe der Seite, dann die gerade und letztlich die ungerade Prüfsumme vorangestellt. Wenn Sie das berücksichtigen, können Sie nach Herzenslust ändern und erweitern, ein Maschinensprachemonitor und Assemblerkenntnisse vorausgesetzt.

Probieren Sie mal einige Routinen aus

Wenn Sie das Programm genauso wie abgebildet im Speicher stehen haben, so können Sie durch »sys 49161« die Eliminierung aller REMs erzielen. Die Routine startet mit der Meldung »20779 Bytes PRG-Laenge« und endet nach 22 Sekunden mit »20019 Bytes PRG-Laenge«. 760 Bytes haben Sie gewonnen oder verloren, wie man's nimmt.

Doch Vorsicht! Wenn Zeilen mit einem REM beginnen, wird die ganze Zeile entfernt. Sprünge zu dieser Zeile werden nicht berücksichtigt. Wenn das Programm mit der Fehlermeldung »undef'd statement« unterbricht, korrigieren Sie einfach den Sprungbefehl durch die Zeile, welcher der entfernten unmittelbar folgt.

Wenn Sie allerdings schon vorher überblicken, welche Zeilen »fällig« sein werden, können Sie durch zum Beispiel »sys 49167,"2000"« alle Zeilen mit dem Wortlaut "2000" ausfindig machen und korrigieren.

»sys 49155« kommt nach etwa 13 Minuten mit »17828 Bytes PRG-Laenge« zurück. Diese lange Zeit rührt davon her, daß nach jedem eliminiertem Space alle Programmzeilen wieder gekoppelt werden müssen, bei einem Gewinn von 2191 Bytes bedeutet das 2191 mal koppeln, und das Verschieben selbst braucht natürlich auch noch ein paar Millisekunden. Dann sind schnell, ehe man sich versieht, einige Minuten verflossen.

»sys 49179« als Package-Befehl bringt Ihnen nach 9 Minuten weitere 2095 Bytes Speicherplatzgewinn. Nach diesem 3-Stufen-Plan haben Sie insgesamt 5046 Bytes, die für die Funktion des Programms entbehrlich sind, entfernt. Durch den Platzgewinn werden Ihre Programme übrigens schneller, da erstens die Strings mehr Platz haben und seltener aufgeräumt werden müssen und zweitens weniger Zeilenübergänge stattfinden.

Ich möchte nochmals betonen, daß sich derartige »Rumpfpakete« ausschließlich für den RUN-Modus eignen. Listen, Korrigieren und so weiter sollte immer an einem »gesunden« Programm stattfinden.

(Johannes Mockenhaupt / ev)

Bild 1: Mit folgendem Text meldet sich das Menü nach »sys 49152«

sys49155:	Space-Killer
sys49158:	Space-Expander
sys49161:	REM-Killer
sys49164:	Zeilen löschen
sys49167:	Text suchen
sys49170:	Colon-Maker
sys49173:	Space-Maker
sys49179:	Package
sys49182:	CPU-Register ein
sys49185:	CPU-Register aus
sys49188:	BASIC-Ende setzen
sys49191:	Variablen dumpen
sys49194:	ROM=> RAM
sys49197:	Zeichensatz nach $7000
sys49200:	Zeichensatz ändern
sys49203:	Zeichensatz speichern
sys49206:	Zeichensatz, alt

Im Züge technisch-wirtschaftlicher Überlegungen...!

Listing »Super-Utilities«

```
100 REM    *************************          <227>
110 REM    *                       *          <081>
120 REM "[SPACE]*[SPACE]JOHANNES[SPACE]
   MOCKENHAUPT[SPACE]*                        <018>
130 REM    *                       *          <101>
140 REM "[SPACE]*[SPACE2]K£LN,[SPACE]AUGUST
   [SPACE]1984[SPACE3]*                       <173>
150 REM    *                       *          <121>
160 REM    *                       *          <131>
170 REM "[SPACE]*[SPACE2]HILFSROUTINEN[SPACE]F]R
   :[SPACE2]*                                 <072>
180 REM    *                       *          <151>
190 REM "[SPACE]*[SPACE4]COMMODORE[SPACE]C'64
   [SPACE4]*                                  <060>
200 REM    *                       *          <171>
210 REM    *************************          <081>
220 :                                         <022>
230 PRINT "[CLEAR]DIE[SPACE]DATEN[SPACE]WERDEN
   [SPACE]SEITENWEISE                         <124>
240 PRINT "[DOWN]GEPOKED[SPACE]UND[SPACE]
   KONTROLLIERT.[DOWN]                        <138>
250 N=12*2↑12:          REM "[SPACE]$C000     <046>
260 : FOR J=0 TO 15                           <246>
270 :     READ A$,PG,PU:REM "[SPACE]PR]FSUMMEN
                    <078>
280 :          PRINT "SEITE[SPACE]"A$":";     <072>
290 :            FOR I=0 TO 127               <071>
300 :              READ A,B                   <157>
310 :                PG=PG-A:PU=PU-B          <096>
320 :                POKE N,A:POKE N+1,B      <053>
330 :              N=N+2                      <175>
340 :            NEXT I                       <090>
350 :     IF PG=0 AND PU=0 THEN  370          <020>
360 :        PRINT "[RVSON]DATA-FEHLER[RVOFF]!"
   :STOP                                      <140>
370 :     PRINT "IN[SPACE]ORDNUNG             <156>
380 : NEXT J                                  <131>
390 :                                         <193>
400 PRINT "[CLEAR]ALLE[SPACE]PRUEFSUMMEN[SPACE]
   STIMMEN,[SPACE]DURCH                       <019>
410 PRINT "[DOWN]WAHL[SPACE]DER[SPACE]
   GERAETENUMMER[SPACE](1[SPACE]ODER[SPACE]8)
                    <035>
420 PRINT "[DOWN]KOENNEN[SPACE]SIE[SPACE]DEN
   [SPACE]$C-BLOCK[SPACE]DIREKT[SPACE]ALS     <217>
430 PRINT "[DOWN]MASCHINEN-PRG[SPACE]ABSOLUT
   [SPACE]SPEICHERN.                          <016>
440 PRINT "[DOWN]KEINE[SPACE]EINGABE
   :[SPACE]ENDE.                              <224>
450 INPUT "[DOWN2]GERAETENUMMER:[SPACE]";GN
                    <009>
460 IF GN<>1 AND GN<>8 THEN 480               <165>
470 SYS 53138 "HILFSROUT.[SPACE]EXE",GN       <078>
480 PRINT "[DOWN]MIT[SPACE]'SYS[SPACE]49152'
   [SPACE]BEKOMMEN[SPACE]SIE[SPACE]EIN[SPACE]
   MENUE.                                     <028>
1000 DATA C0,12017,12700:REM $C000-C0FF       <026>
1010 :                                        <047>
1020 DATA 76, 57,192, 76, 71,194, 76, 14      <112>
1030 DATA 195, 76,210,195, 76, 96,196, 76     <232>
1040 DATA 232,196, 76, 97,197, 76,101,197     <032>
1050 DATA 76,148,197, 76,192,199, 76, 66      <005>
1060 DATA 201, 76, 48,202, 76, 88,194, 76     <199>
1070 DATA 57,202, 76, 22,203, 76,109,203      <235>
1080 DATA 76, 17,204, 76, 90,207, 76, 26      <162>
1090 DATA 207, 32, 14,194,160,  0,169, 94     <210>
1100 DATA 162,192,133,251,134,252,177,251     <211>
1110 DATA 208, 10,169, 14, 32, 22,231,169     <016>
1120 DATA  8, 76, 22,231, 32,210,255,200      <224>
1130 DATA 208,236,230,252,208,232, 83, 89     <149>
1140 DATA 83, 52, 57, 49, 53, 53, 58, 32      <124>
1150 DATA 211, 80, 65, 67, 69, 45,203, 73     <228>
1160 DATA 76, 76, 69, 82, 13, 83, 89, 83      <161>
1170 DATA 52, 57, 49, 53, 56, 58, 32,211      <198>
1180 DATA 80, 65, 67, 69, 45,197, 88, 80      <232>
1190 DATA 65, 78, 68, 69, 82, 13, 83, 89      <194>
1200 DATA 83, 52, 57, 49, 54, 49, 58, 32      <190>
1210 DATA 210,197,205, 45,203, 73, 76, 76     <130>
1220 DATA 69, 82, 13, 83, 89, 83, 52, 57      <214>
1230 DATA 49, 54, 52, 58, 32,218, 69, 73      <012>
1240 DATA 76, 69, 78, 32, 76, 79, 69, 83      <250>
1250 DATA 67, 72, 69, 78, 13, 83, 89, 83      <252>
1260 DATA 52, 57, 49, 54, 55, 58, 32,212      <033>
1270 DATA 69, 88, 84, 32, 83, 85, 67, 72      <015>
1280 DATA 69, 78, 13, 83, 89, 83, 52, 57      <024>
1290 DATA 49, 55, 48, 58, 32,195, 79, 76      <087>
1300 DATA 79, 78, 45,205, 65, 75, 69, 82      <095>
1310 DATA 13, 83, 89, 83, 52, 57, 49, 55      <047>
1320 DATA 51, 58, 32,211, 80, 65, 67, 69      <095>
1330 DATA 45,205, 65, 75, 69, 82, 13, 83      <109>
1340 :                                        <123>
1350 :                                        <133>
1360 DATA C1,9851,9900:REM $C100-C1FF         <059>
1370 :                                        <153>
1380 DATA 89, 83, 52, 57, 49, 55, 57, 58      <127>
1390 DATA 32, 80, 65, 67, 75, 65, 71, 69      <125>
1400 DATA 13, 83, 89, 83, 52, 57, 49, 56      <138>
1410 DATA 50, 58, 32,195,208,213, 45,210      <064>
1420 DATA 69, 71, 73, 83, 84, 69, 82, 32      <158>
1430 DATA 69, 73, 78, 13, 83, 89, 83, 52      <172>
1440 DATA 57, 49, 56, 53, 58, 32,195,208      <019>
1450 DATA 213, 45,210, 69, 71, 73, 83, 84     <016>
1460 DATA 69, 82, 32, 65, 85, 83, 13, 83      <192>
1470 DATA 89, 83, 52, 57, 49, 56, 56, 58      <217>
1480 DATA 32,194,193,211,201,195, 45,197      <244>
1490 DATA 78, 68, 69, 32, 83, 69, 84, 90      <238>
1500 DATA 69, 78, 13, 83, 89, 83, 52, 57      <244>
1510 DATA 49, 57, 49, 58, 32,214, 65, 82      <038>
1520 DATA 73, 65, 66, 76, 69, 78, 32, 68      <011>
1530 DATA 85, 77, 80, 69, 78, 13, 83, 89      <023>
1540 DATA 83, 52, 57, 49, 57, 52, 58, 32      <017>
1550 DATA 210,207,205, 32, 61, 62, 32,210     <186>
1560 DATA 193,205, 13, 83, 89, 83, 52, 57     <135>
1570 DATA 49, 57, 55, 58, 32,218, 69, 73      <104>
1580 DATA 67, 72, 69, 78, 83, 65, 84, 90      <072>
1590 DATA 32, 78, 65, 67, 72, 32, 36, 55      <064>
1600 DATA 48, 48, 48, 13, 83, 89, 83, 52      <083>
1610 DATA 57, 50, 48, 48, 58, 32,218, 69      <140>
1620 DATA 73, 67, 72, 69, 78, 83, 65, 84      <113>
1630 DATA 90, 32, 65, 69, 78, 68, 69, 82      <121>
1640 DATA 78, 13, 83, 89, 83, 52, 57, 50      <119>
1650 DATA 48, 51, 58, 32,218, 69, 73, 67      <180>
1660 DATA 72, 69, 78, 83, 65, 84, 90, 32      <144>
1670 DATA 83, 80, 69, 73, 67, 72, 69, 82      <158>
1680 DATA 78, 13, 83, 89, 83, 52, 57, 50      <159>
1690 DATA 48, 54, 58, 32,218, 69, 73, 67      <223>
1700 :                                        <228>
1710 :                                        <238>
1720 DATA C2,17103,16952:REM $C200-C2FF       <001>
1730 :                                        <002>
1740 DATA 72, 69, 78, 83, 65, 84, 90, 44      <227>
1750 DATA 32, 65, 76, 84, 13,  0,169, 13      <212>
1760 DATA 76,210,255, 32, 14,194,141,  0      <103>
1770 DATA  4, 56,165, 45,229, 43,170,165      <131>
1780 DATA 46,229, 44, 32,205,189,162,  0      <135>
1790 DATA 189, 52,194,208,  1, 96, 32,210     <148>
1800 DATA 255,232,208,244, 32,194, 89, 84     <013>
1810 DATA 69, 83, 32,208,210,199, 45,204      <222>
1820 DATA 65, 69, 78, 71, 69, 13,  0, 32      <243>

1830 DATA 19,194,166, 43,165, 44,134,251      <036>
1840 DATA 133,252,160,  1,177,251,208, 41     <028>
1850 DATA 32, 51,165,166, 43,165, 44,134      <255>
1860 DATA 45,133, 46,160,  0,177, 45,170      <211>
1870 DATA 200,177, 45,208,242,165, 45, 24     <067>
1880 DATA 105,  2,133, 45,144,  2,230, 46     <168>
1890 DATA 32, 96,166, 32, 19,194, 76,123      <001>
1900 DATA 227,160,  3,200,177,251,240, 90     <087>
1910 DATA 201, 34,208,  9,200,177,251,240     <096>
1920 DATA 81,201, 34,208,247,201, 32,208      <106>
1930 DATA 234,166,251,165,252,134,141,133     <016>
1940 DATA 142, 24,152,101,251,133,251,144     <217>
1950 DATA  2,230,252,160,  0,200,177,251      <076>
1960 DATA 240,  4,201, 32,240,247, 24,152     <089>
1970 DATA 101,251,133,253,165,252,105,  0     <198>
1980 DATA 133,254,160,  0,177,253,145,251     <222>
1990 DATA 200,208,249,230,252,230,254,165     <073>
2000 DATA 46,197,254,176,239,238,  0,216      <214>
2010 DATA 32, 51,165,166,141,165,142, 76      <210>
2020 DATA 78,194,160,  0,177,251,170,200      <215>
2030 DATA 177,251, 76, 78,194,170,192,  5     <198>
2040 DATA 176,  3, 76, 36,195,136,169, 32     <155>
2050 DATA 209,251,240,  4,200, 76,117,195     <245>
2060 :                                        <078>
2070 :                                        <088>
2080 DATA C3,20694,18471:REM $C300-C3FF       <085>
2090 :                                        <108>
2100 DATA 200,138, 76, 88,195,169,199, 72     <073>
2110 DATA 169, 60, 72, 76,115,  0, 32, 19     <111>
2120 DATA 194,166, 43,165, 44,134,251,133     <116>
2130 DATA 252,160,  1,177,251,208,  3, 76     <227>
2140 DATA 88,194,160,  3,200,177,251,208      <086>
```

Listing »Super-Utilities« (Fortsetzung)

```
2150 DATA  3, 76,199,195,201, 34,208,  9    <212>
2160 DATA 200,177,251,240,244,201, 34,208   <188>
2170 DATA 247,170, 16,232,201,163,240,228   <203>
2180 DATA 201,166,240,224,201,164,240,165   <001>
2190 DATA 201,167,240,161,201,169,240,157   <018>
2200 DATA 201,175,240,153,201,176,240,149   <027>
2210 DATA 201,180,176, 12,201,170,144,  8   <138>
2220 DATA 201,175,240,  4,201,176,208,188   <209>
2230 DATA 200,177,251,240, 90,201, 58,240   <211>
2240 DATA 179,201, 32,240,175,166,251,165   <028>
2250 DATA 252,134,141,133,142, 24,136,152   <022>
2260 DATA 101,251,133,251,144,  2,230,252   <229>
2270 DATA 24,166, 56,202,202,134,254,134    <254>
2280 DATA 37,165,251,133, 36,105,  1,133    <163>
2290 DATA 253,144,  2,230,254,160,255,177   <024>
2300 DATA 36,145,253,136,208,249,177, 36    <051>
2310 DATA 145,253,198, 37,198,254,165, 37   <073>
2320 DATA 197,252,176,233,160,  1,169, 32   <013>
2330 DATA 145,251,238,  0,216, 32, 51,165   <217>
2340 DATA 166,141,165,142, 76, 21,195,160   <080>
2350 DATA  0,177,251,170,200,177,251, 76    <035>
2360 DATA 21,195, 32, 19,194, 32, 51,165    <206>
2370 DATA 166, 43,165, 44,134,251,133,252   <106>
2380 DATA 160,  1,177,251,208,  3, 76, 88   <181>
2390 DATA 194,160,  3,200,177,251,240, 54   <070>
2400 DATA 201, 34,208,  9,200,177,251,240   <076>
2410 DATA 45,201, 34,208,247,201,143,208    <137>
2420 :                                      <183>
2430 :                                      <193>
2440 DATA C4,20508,20440:REM $C400-C4FF     <208>
2450 :                                      <213>
2460 DATA 234,192,  5,176, 44,160,  0,177   <050>
2470 DATA 251,133,253,200,177,251,133,254   <042>
2480 DATA 160,  0,177,253,145,251,200,208   <206>
2490 DATA 249,230,252,230,254,165, 46,197   <030>
2500 DATA 254,176,239, 76,213,195,160,  0   <196>
2510 DATA 177,251,170,200,177,251, 76,220   <039>
2520 DATA 195,132,141,160,  0,177,251,133   <248>
2530 DATA 253,200,177,251,133,254,164,141   <104>
2540 DATA 192,  5,144,204,136,177,251,201   <016>
2550 DATA 58,240,245,201, 32,240,241,200    <011>
2560 DATA 169,  0,145,251, 56,152,101,251   <240>
2570 DATA 133,251,144,180,230,252,176,176   <150>
2580 DATA 32,223,196,132,253,133,254, 32    <056>
2590 DATA 223,196,132,141,133,142, 32, 19   <063>
2600 DATA 194,166, 43,165, 44,134,251,133   <086>
2610 DATA 252,160,  0,177,251,170,200,177   <083>
2620 DATA 251,240, 89,208,  6,160,  1,177   <211>
2630 DATA 251,208,234,160,  3,177,251,197   <116>
2640 DATA 254,144,242,208,  7,136,177,251   <127>
2650 DATA 197,253,144,233,166,251,165,252   <241>
2660 DATA 134,139,133,140,160,  0,177,139   <137>
2670 DATA 170,200,177,139,208,  8,240, 25   <100>
2680 DATA 160,  1,177,139,208,234,160,  3   <058>
2690 DATA 177,139,197,142,144,242,208,  9   <187>
2700 DATA 136,177,139,197,141,144,233,240   <033>
2710 DATA 231,160,  0,177,139,145,251,200   <180>
2720 DATA 208,249,230,252,230,140,165, 46   <248>
2730 DATA 197,140,176,239, 76, 88,194, 32   <189>
2740 DATA 253,174, 32,138,173, 76,247,183   <237>
2750 DATA 32,253,174,162,  0, 32,115,  0    <015>
2760 DATA 240, 10, 72,104,157, 16,  2,232   <075>
2770 DATA 224, 48,144,241,169,  0,157, 16   <155>
2780 :                                      <032>
2790 :                                      <042>
2800 DATA C5,17505,17163:REM $C500-C5FF     <071>
2810 :                                      <062>
2820 DATA  2,166, 43,165, 44,134,251,133    <199>
2830 DATA 252,160,  1,177,251,208, 13,104   <251>
2840 DATA 72, 96,160,  0,177,251,170,200    <218>
2850 DATA 177,251,208,233,160,  3,200,177   <072>
2860 DATA 251,240,239,205, 16,  2,208,246   <032>
2870 DATA 24,152,132,141,101,251,133,253    <125>
2880 DATA 165,252,105,  0,133,254,160,  0   <246>
2890 DATA 200,185, 16,  2,240, 17,201, 63   <207>
2900 DATA 240,246,177,253, 72,104,217, 16   <128>
2910 DATA  2,240,237,164,141,208,207,160    <127>
2920 DATA  2,177,251,170,200,177,251, 32    <089>
2930 DATA 205,189,169, 13, 32,210,255,208   <159>
2940 DATA 177,169, 32,208,  2,169, 58,133   <082>
2950 DATA 253,166, 43,165, 44,134,251,133   <177>
2960 DATA 252,160,  1,177,251,208,  3, 76   <036>
2970 DATA 88,194,160,  4,177,251,197,253    <167>
2980 DATA 208,  4, 73, 26,145,251,160,  0   <255>
2990 DATA 177,251,170,200,177,251, 76,109   <015>
3000 DATA 197, 76, 88,194, 32, 19,194, 32   <104>
3010 DATA  5,195,134, 65,133, 66, 32,158    <097>
3020 DATA 199, 32,147,199,134, 59,133, 60   <216>
3030 DATA 228, 65,229, 66,176,  3, 76, 72   <079>
3040 DATA 178, 32,147,199,134, 63,133, 64   <232>
3050 DATA 160,  1,177, 67,240,211,169,127   <228>
3060 DATA 133, 21,133, 20, 32,  5,195, 32   <071>
3070 DATA 142,166,176, 47,160,  0,177,139   <000>
3080 DATA 170,200,177,251,240,187, 72,200   <093>
3090 DATA 165, 59,145, 34,200,165, 60,177   <019>
3100 DATA 36,134, 67,104,133, 68,165, 59    <241>
3110 DATA 24,101, 63,133, 59,165, 60,101    <227>
3120 DATA 64,133, 60,144,215,230,122,208    <079>
3130 DATA  2,230,123,160,  0,177,122,208    <239>
3140 :                                      <138>
3150 :                                      <148>
3160 DATA C6,13653,16163:REM $C600-C6FF     <179>
3170 :                                      <168>
3180 DATA 19,160,  2,177,122,240,197,165    <105>
3190 DATA 122, 24,105,  5,133,122,144,235   <094>
3200 DATA 230,123,176,231,201, 34,208, 11   <152>
3210 DATA 32,115,  0,201,  0,240,226,201    <255>
3220 DATA 34,208,245,201,137,240, 23,201    <175>
3230 DATA 138,240, 19,201,141,240, 15,201   <181>
3240 DATA 167,240, 11,201,203,208,190, 32   <194>
3250 DATA 115,  0,201,164,208,191,238,  0   <108>
3260 DATA 216, 32,115,  0,176,183,132, 20   <118>
3270 DATA 132, 21,144, 49,170,165, 21,133   <182>
3280 DATA 34,201, 25,176,168,165, 20, 10    <144>
3290 DATA 69, 34, 10,165, 34,101, 20,133    <098>
3300 DATA 20,165, 34,101, 21,133, 21,  6    <049>
3310 DATA 20,133, 21,138,101, 20,133, 20    <150>
3320 DATA 144,  2,230, 21,200,177,122,201   <215>
3330 DATA 32,240,249,201, 58,165, 20,197    <255>
3340 DATA 65,165, 21,229, 66,144, 72,132    <219>
3350 DATA 206, 32, 60,199, 56,162,144, 32   <226>
3360 DATA 73,188, 32,221,189,160,255,200    <048>
3370 DATA 185,  0,  1,208,250,165,122,166   <236>
3380 DATA 123,133, 95,134, 96, 56,152,229   <058>
3390 DATA 206, 48, 38,240,106,133,206,165   <105>
3400 DATA 45,133, 90, 24,101,206,133, 88    <012>
3410 DATA 165, 46,133, 91,105,  0,133, 89   <235>
3420 DATA 197, 56,144,  6,165, 88,197, 55   <018>
3430 DATA 176,109, 32,191,163,240, 72,144   <148>
3440 DATA 90, 73,255,168,200,162,  0,165    <057>
3450 DATA 95,197, 45,165, 96,229, 46,176    <102>
3460 DATA 54,147, 17, 17, 17, 17, 33, 33    <185>
3470 DATA 203, 80, 73, 66, 79, 79, 70, 84   <216>
3480 DATA 33,206, 80, 68, 76, 70, 79, 73    <222>
3490 DATA 66, 86, 81, 85, 13, 17, 68, 33    <182>
3500 :                                      <243>
3510 :                                      <253>
3520 DATA C7,15307,15592:REM $C700-C7FF     <034>
3530 :                                      <017>
3540 DATA 50, 58, 57, 53, 33,204, 80, 70    <009>
3550 DATA 77, 79, 17, 29, 29, 13,  0,230    <231>
3560 DATA 95,208,196,230, 96,208,192,160    <037>
3570 DATA  0,185,  0,  1,240,  5,145,122    <223>
3580 DATA 200,208,246, 32,158,199, 32,115   <042>
3590 DATA  0,144,251,160,  0,201, 44,208    <136>
3600 DATA  2,169,137,170, 76,255,197,162    <029>
3610 DATA 16, 76, 58,164,  2,160,  0,185    <082>
3620 DATA 225,198,240, 24, 72, 41,127,201   <027>
3630 DATA 32,144,  6,104,233,  1, 76, 82    <092>
3640 DATA 199,104, 32,210,255,200,208,231   <139>
3650 DATA 133, 34,134, 35,160,  2,132,211   <248>
3660 DATA 32,123,164,177, 34,197, 20,200    <066>
3670 DATA 177, 34,229, 21,176, 81,160,  0   <239>
3680 DATA 177, 34,170,200,177, 34,134, 34   <090>
3690 DATA 133, 35,177, 34,240, 65,165, 99   <067>
3700 DATA 101, 63,133, 99,165, 98,101, 64   <070>
3710 DATA 133, 98,176,  4,201,252,144,204   <120>
3720 DATA 76, 72,178, 32,253,174, 32,107    <091>
3730 DATA 169,166, 20,165, 21, 96, 32, 51   <047>
3740 DATA 165, 24,165, 34,105,  2,133, 45   <046>
3750 DATA 165, 35,105,  0,133, 46,165, 65   <061>
3760 DATA 166, 66,133, 20,134, 21, 32, 19   <065>
3770 DATA 166,165, 95,133, 67,134, 68, 96   <159>
3780 DATA 32, 19,194,166, 43,165, 44,134    <150>
3790 DATA 251,133,252,238,  0,216,138, 24   <195>
3800 DATA 105,  3,133,247,165,252,105,  0   <149>
3810 DATA 133,248,160,  1,177,251,208,  3   <168>
3820 DATA 76, 88,194,160,  0,132,  2,230    <080>
3830 DATA 247,208,  2,230,248,177,247,240   <041>
3840 DATA 55,201, 34,208, 29,230,  2,230    <139>
```

Listing »Super-Utilities« (Fortsetzung)

```
3850 DATA 247,208,  2,230,248,177,247,240  <062>
3860 :                                     <093>
3870 :                                     <103>
3880 DATA C8,17613,21819:REM $C800-C8FF    <144>
3890 :                                     <123>
3900 DATA 39,201, 34,240,224,208,240,160   <093>
3910 DATA  0,177,251,170,200,177,251, 76   <065>
3920 DATA 199,199,201,137,240,241,201,138  <230>
3930 DATA 240,237,201,142,240,233,201,143  <213>
3940 DATA 240,229,201,167,208,191,240,223  <240>
3950 DATA 160,  0,177,251,133,253,200,177  <148>
3960 DATA 251,133,254,200,177,253,133,249  <009>
3970 DATA 200,177,253,133,250, 56,160,  0  <115>
3980 DATA 177,253,229,251,200,177,253,229  <040>
3990 DATA 252,208,188,166, 43,165, 44,134  <208>
4000 DATA 247,133,248,160,  1,177,247,208  <214>
4010 DATA  3, 76,183,200, 24,165,247,105   <115>
4020 DATA  3,133,122,165,248,105,  0,133   <112>
4030 DATA 123,160,  0,230,122,208,  2,230  <108>
4040 DATA 123,177,122,240, 55,201, 34,208  <236>
4050 DATA 14,230,122,208,  2,230,123,177   <189>
4060 DATA 122,240, 41,201, 34,208,242,201  <241>
4070 DATA 137,240, 12,201,138,240,  8,201  <208>
4080 DATA 141,240,  4,201,167,208,212, 32  <219>
4090 DATA 115,  0, 32,107,169,165, 21,197  <196>
4100 DATA 250,208,198,165, 20,197,249,208  <118>
4110 DATA 192, 76,  7,200,160,  0,177,247  <170>
4120 DATA 170,200,177,247, 76, 79,200,238  <081>
4130 DATA  0,  4,160,  0, 24,165,253,105   <070>
4140 DATA 255,133,139,165,254,105,255,133  <195>
4150 DATA 140,165,  2,240,  4,169, 34, 16  <152>
4160 DATA  2,169, 58,145,139,200,201, 34   <009>
4170 DATA 240,247, 24,165,253,105,  4,133  <065>
4180 DATA 36,165,254,105,  0,133, 37,230   <023>
4190 DATA 139,208,  2,230,140,164,  2,240  <027>
4200 DATA  4,198,  2,240,242,177, 36,145   <007>
4210 DATA 139,200,208,249,230,140,230, 37  <202>
4220 :                                     <198>
4230 :                                     <208>
4240 DATA C9,13954,13099:REM $C900-C9FF    <001>
4250 :                                     <228>
4260 DATA 165, 46,197, 37,176,239,165,251  <236>
4270 DATA 105,  4,133,139,165,252,105,  0  <110>
4280 DATA 133,140,177,139,240, 10,230,139  <016>
4290 DATA 208,248,230,140,208,244,240,242  <077>
4300 DATA 160,  0, 24,165,139,105,  1,145  <090>
4310 DATA 251,200,165,140,105,  0,145,251  <240>
4320 DATA 200,177,139,208,  3, 76, 88,194  <189>
4330 DATA 32, 51,165,166,251,165,252, 76   <239>
4340 DATA 199,199,173, 20,  3,172, 21,  3  <097>
4350 DATA 141, 13,202,140, 14,202,169, 91  <235>
4360 DATA 160,201,120,141, 20,  3,140, 21  <176>
4370 DATA  3, 88, 96,160,  0,185,123,201   <122>
4380 DATA 240, 66, 41, 63,153,  0,  4,169  <081>
4390 DATA 99,153, 80,  4,173,134,  2,153   <147>
4400 DATA  0,216,153, 40,216,153, 80,216   <241>
4410 DATA 200,208,226, 73, 82, 81, 32, 32  <207>
4420 DATA 66, 82, 75, 32, 32, 78, 77, 73   <095>
4430 DATA 32, 32, 32, 80, 67, 32, 32, 83   <079>
4440 DATA 86, 32, 66, 68, 73, 90, 67, 32   <114>
4450 DATA 65, 67, 32, 88, 82, 32, 89, 82   <129>
4460 DATA 32, 83, 80,  0,173, 14,202, 32   <149>
4470 DATA 22,202,173, 13,202, 32, 22,202   <038>
4480 DATA 173, 23,  3, 32, 15,202,173, 22  <217>
4490 DATA  3, 32, 22,202,173, 25,  3, 32   <124>
4500 DATA 15,202,173, 24,  3, 32, 22,202   <231>
4510 DATA 189,  6,  1, 32, 15,202,189,  5  <165>
4520 DATA  1, 32, 22,202,169, 32, 32, 43   <207>
4530 DATA 202,169,  8,133,164,189,  4,  1  <031>
4540 DATA 133,165,  6,165,144,  4,169, 42  <092>
4550 DATA 208,  2,169, 46, 32, 43,202,198  <102>
4560 DATA 164,208,239,189,  3,  1, 32, 15  <060>
4570 DATA 202,189,  2,  1, 32, 15,202,189  <060>
4580 :                                     <047>
4590 :                                     <057>
4600 DATA CA,16040,17390:REM $CA00-CAFF    <117>
4610 :                                     <078>
4620 DATA  1,  1, 32, 15,202,138, 24,105   <046>
4630 DATA  6, 32, 15,202, 76, 49,234, 72   <080>
4640 DATA 169, 32, 32, 43,202,104, 72, 74  <151>
4650 DATA 74, 74, 74, 32, 31,202,104, 41   <139>
4660 DATA 15,201, 10,144,  4,233,  9,208   <143>
4670 DATA  2,  9, 48,153,  0,  4,200, 96   <220>
4680 DATA 173, 13,202,172, 14,202, 76, 82  <014>
4690 DATA 201,166, 45,165, 46,134,251,133  <129>
4700 DATA 252, 56,165,251,229, 47,165,252  <154>
4710 DATA 229, 48,176, 85,160,  0,132,211  <052>
4720 DATA 177,251,133,253, 41,127,170,200  <202>
4730 DATA 177,251,133,254, 41,127,168, 32  <175>
4740 DATA 236,202,165,253,  5,254, 48, 58  <091>
4750 DATA 162,  0, 32,246,202, 24,165,251  <079>
4760 DATA 105,  2,133, 34,165,252,105,  0  <034>
4770 DATA 133, 35, 32,166,187, 32,221,189  <168>
4780 DATA 32, 30,171,169, 13, 32,210,255   <110>
4790 DATA 173,141,  2,240,  8,201,  1,240  <010>
4800 DATA 247,201,  2,240, 12, 24,165,251  <126>
4810 DATA 105,  7,170,165,252,105,  0,144  <140>
4820 DATA 156, 96,165,253, 48, 42,162, 36  <178>
4830 DATA 160, 61, 32,236,202,169, 34, 32  <165>
4840 DATA 210,255,160,  4,177,251,133,254  <020>
4850 DATA 136,177,251,133,253,136,177,251  <142>
4860 DATA 133,141,160,  0,196,141,176,187  <045>
4870 DATA 177,253, 32,210,255,200,208,244  <099>
4880 DATA 165,254, 16,180,162, 37, 32,246  <021>
4890 DATA 202, 24,165,251,105,  2,133,100  <002>
4900 DATA 165,252,105,  0,133,101, 32, 97  <231>
4910 DATA 175, 76,125,202,138,240,  3, 32  <248>
4920 DATA 210,255,152, 76,210,255,160, 61  <099>
4930 DATA 208,242,194,193,211,201,195, 32  <164>
4940 :                                     <153>
4950 :                                     <163>
4960 DATA CB,16618,16505:REM $CB00-CBFF    <234>
4970 :                                     <183>
4980 DATA 43, 32,203,197,210,206,193,204   <158>
4990 DATA 32, 74, 69, 84, 90, 84, 32, 73   <151>
5000 DATA 77, 32,210,193,205,  0,160,  0   <227>
5010 DATA 169,160,132,251,133,252,169,224  <041>
5020 DATA 132,253,133,254,162, 32,177,251  <253>
5030 DATA 145,251,177,253,145,253,200,208  <059>
5040 DATA 245,230,252,230,254,202,208,238  <062>
5050 DATA 185,250,202,240,  6, 32,210,255  <173>
5060 DATA 200,208,245, 96,208,210,199, 45  <253>
5070 DATA 203, 79, 76, 76, 73, 83, 73, 79  <038>
5080 DATA 78, 44, 32,193, 69, 78, 68, 69   <054>
5090 DATA 82, 85, 78, 71, 32, 78, 73, 67   <005>
5100 DATA 72, 84, 32, 77, 79, 69, 71, 76   <018>
5110 DATA 73, 67, 72, 33,  0,169,108,160   <102>
5120 DATA  0,197, 46,176, 11,185, 68,203   <167>
5130 DATA 240,201, 32,210,255,200,208,245  <089>
5140 DATA 132, 55,197, 56,176,  2,133, 56  <189>
5150 DATA 162, 16,169,208,132,251,133,252  <129>
5160 DATA 169,112,132,253,133,254,173, 14  <137>
5170 DATA 220, 41,254,141, 14,220,165,  1  <237>
5180 DATA 41,251,133,  1,177,251,145,253   <055>
5190 DATA 200,208,249,230,252,230,254,202  <206>
5200 DATA 208,242,165,  1,  9,  4,133,  1  <079>
5210 DATA 173, 14,220,  9,  1,141, 14,220  <178>
5220 DATA 173,  0,221, 41,252,  9,  2,141  <142>
5230 DATA  0,221,169,188,141, 24,208,169   <117>
5240 DATA 108,141,136,  2,185,223,203,240  <158>
5250 DATA 53, 32,210,255,200,208,245,147   <172>
5260 DATA 17, 17, 17, 14, 36, 54, 67, 48   <159>
5270 DATA 48, 58,194, 73, 76, 68, 83, 67   <245>
5280 DATA 72, 73, 82, 77, 13, 17, 36, 55   <181>
5290 DATA 48, 48, 48, 58,218, 69, 73, 67   <007>
5300 :                                     <002>
5310 :                                     <012>
5320 DATA CC,12980,12282:REM $CC00-CCFF    <082>
5330 :                                     <032>
5340 DATA 72, 69, 78, 71, 69, 78, 69, 82   <013>
5350 DATA 65, 84, 79, 82, 13,  0, 76, 88   <211>
5360 DATA 194,169,147, 32,210,255,160,  0  <240>
5370 DATA 185, 38,204,208,  3, 76,166,204  <209>
5380 DATA 32,210,255,200,208,242, 19, 17   <250>
5390 DATA 17, 17, 77, 73, 84, 32, 18,195   <091>
5400 DATA 213,210,211,207,210,146, 32, 32  <050>
5410 DATA 32,218, 69, 73, 67, 72, 69, 78   <123>
5420 DATA 32, 69, 73, 78, 83, 84, 69, 76   <088>

5430 DATA 76, 69, 78, 44, 13, 77, 73, 84   <093>
5440 DATA 32, 18,210,197,212,213,210,206   <100>
5450 DATA 146, 32, 32, 32, 87, 65, 69, 72  <146>
5460 DATA 76, 69, 78, 46, 13, 29, 29, 29   <122>
5470 DATA 29, 18, 70, 55,146, 58, 32, 32   <161>
5480 DATA 32, 32, 32, 32,218, 69, 73, 67   <169>
5490 DATA 72, 69, 78, 83, 65, 84, 90, 13   <148>
5500 DATA 29, 29, 29, 29, 18,210,213,206   <030>
5510 DATA 47,211,212,207,208,146, 58,193   <190>
5520 DATA 66, 66, 82, 85, 67, 72, 46, 17   <179>
5530 DATA 13, 29, 29, 29, 29, 29, 29, 29   <183>
5540 DATA 29, 29, 29, 29, 29,  0,168,132   <236>
5550 DATA 141,152,201, 32,176,  4,169, 32  <119>
```

```
5560 DATA 208,  7,170, 16,  4,201,160,144  <075>
5570 DATA 245, 32,210,255,230,141,165,141  <026>
5580 DATA 41, 15,208,  9,169, 13, 32,210   <050>
5590 DATA 255,169, 11,133,211,200,208,217  <049>
5600 DATA 185,219,204,240, 27, 32,210,255  <015>
5610 DATA 200,208,245, 19, 17, 17, 17, 17  <135>
5620 DATA 17, 17, 17, 17, 29, 29, 29, 29   <012>
5630 DATA 29, 29, 29, 29, 29, 29, 29,  0   <231>
5640 DATA 133,251,169,  0,133,252, 32, 65  <212>
5650 DATA 207,201, 29,208, 37,230,252,230  <061>
5660 :                                     <108>
5670 :                                     <118>
5680 DATA CD,16915,17377:REM $CD00-CDFF    <203>
5690 :                                     <138>
5700 DATA 211,166,252,224, 16,144,239,198  <179>
5710 DATA 211,198,252,164,251,192, 15,176  <189>
5720 DATA 229,230,251,169, 13, 32,210,255  <135>
5730 DATA 169,  0,133,252,  9, 11,133,211  <245>
5740 DATA 208,208,201, 19,208,  4,160,  0  <255>
5750 DATA 240,166,201,157,208, 10,166,252  <215>
5760 DATA 240,196,198,252,198,211,208,190  <040>
5770 DATA 201, 17,208, 13,164,251,192, 15  <133>
5780 DATA 176,180, 32,210,255,230,251,208  <243>
5790 DATA 173,201,145,208, 11,164,251,240  <248>
5800 DATA 165, 32,210,255,198,251,208,158  <023>
5810 DATA 201,  3,208,  3, 76, 88,194,201  <032>
5820 DATA 13,240, 15,201,136,208,  8,173   <131>
5830 DATA 24,208, 73,  2,141, 24,208, 76   <050>
5840 DATA 246,204,165,252,164,251,240,235  <103>
5850 DATA 24,105, 16,136,208,250, 72, 41   <163>
5860 DATA 128,133,253,104, 41,127,201, 32  <009>
5870 DATA 144,217,201, 64,144,  2, 41, 63  <132>
5880 DATA 164,253,240,  2,  9, 64,133,251  <148>
5890 DATA 173, 24,208, 41,  7, 74,  9, 14  <016>
5900 DATA 133,252,  6,251, 38,252,  6,251  <170>
5910 DATA 38,252,  6,251, 38,252,169,  7   <145>
5920 DATA 133,141,160,  0,185,194,205,208  <047>
5930 DATA  3, 76, 56,206, 32,210,255,200   <193>
5940 DATA 208,242, 19, 17, 17, 17, 17, 17  <166>
5950 DATA 17, 17, 17, 17, 17, 17, 17, 29   <079>
5960 DATA 29, 29, 29, 29, 29, 29, 29, 29   <110>
5970 DATA 29, 29, 29, 29, 29,176,192,192   <013>
5980 DATA 192,192,192,192,192,192,174, 17  <214>
5990 DATA 157,221, 17,157,221, 17,157,221  <154>
6000 DATA 17,157,221, 17,157,221, 17,157   <119>
6010 DATA 221, 17,157,221, 17,157,221, 17  <121>
6020 :                                     <213>
6030 :                                     <223>
6040 DATA CE,19121,18216:REM $CE00-CEFF    <040>
6050 :                                     <243>
6060 DATA 157,221,157,157,157,157,157,157  <090>
6070 DATA 157,157,157,157,173,192,192,192  <103>
6080 DATA 192,192,192,192,192,189,157,157  <118>
6090 DATA 157,157,157,157,157,157,157,145  <125>
6100 DATA 157,221,145,157,221,145,157,221  <108>
6110 DATA 145,157,221,145,157,221,145,157  <123>
6120 DATA 221,145,157,221,145,157,221,  0  <019>
6130 DATA 165,141, 73,  7,168,177,251,133  <000>
6140 DATA 142,169, 15,133,211,160,  8,169  <001>

6150 DATA 32,  6,142,144,  2,  9, 10, 32   <209>
6160 DATA 210,255,136,208,242,169, 13, 32  <067>
6170 DATA 210,255,198,141, 16,218,185,105  <134>
6180 DATA 206,240, 36, 32,210,255,200,208  <074>
6190 DATA 245, 19, 17, 17, 17, 17, 17, 17  <114>
6200 DATA 17, 17, 17, 17, 17, 17, 17, 29   <074>
6210 DATA 29, 29, 29, 29, 29, 29, 29, 29   <105>
6220 DATA 29, 29, 29, 29, 29, 29,  0,133   <100>
6230 DATA 248,133,247, 32, 65,207,201, 29  <094>
6240 DATA 208, 13,166,247,224,  7,176,243  <109>
6250 DATA 230,247, 32,210,255,208,236,201  <199>
6260 DATA 157,208, 11,166,247,240,228,198  <234>
6270 DATA 247, 32,210,255,208,221,201, 17  <168>
6280 DATA 208, 13,164,248,192,  7,176,211  <147>
6290 DATA 32,210,255,230,248,208,204,201   <235>
6300 DATA 145,208, 11,164,248,240,196,198  <018>
6310 DATA 248, 32,210,255,208,189,201, 32  <219>
6320 DATA 240, 11,201, 42,240, 43,201, 13  <100>
6330 DATA 208,177, 76, 22,204,166,247,224  <251>
6340 DATA  8,176,243, 32,210,255,169,128   <209>
6350 DATA 166,247,240,  2, 74,202,208,252  <209>
6360 DATA 73,255,133,253,164,248,177,251   <078>
6370 DATA 37,253,145,251,230,247, 76,139   <035>
6380 :                                     <062>
6390 :                                     <072>
6400 DATA CF,11190,11261:REM $CF00-CFA6    <119>
6410 :                                     <093>
6420 DATA 206,166,247,224,  8,176,214, 32  <032>
6430 DATA 210,255,169,128,166,247,240,  2  <090>
6440 DATA 74,202,208,252,164,248, 17,251   <099>
6450 DATA 208,224,169,  4,160,151,141,136  <102>
6460 DATA  2,140,  0,221, 32, 91,255,169   <163>
6470 DATA  5, 32,210,255,169,  8, 32,210   <175>
6480 DATA 255,169, 14, 32,210,255,169, 11  <087>
6490 DATA 141, 32,208,141, 33,208, 76, 88  <050>
6500 DATA 194, 70,204, 70,207, 32,228,255  <108>
6510 DATA 240,247, 72,230,204,164,211,177  <211>
6520 DATA 209, 41,127,145,209,104, 96, 32  <128>
6530 DATA 253,174, 32,212,225,162,  1,160  <124>
6540 DATA 128,134,185,202,134,141,169,112  <035>
6550 DATA 133,142,169,141, 76,216,255, 32  <209>
6560 DATA 32,202, 79, 72, 65, 78, 78, 69   <250>
6570 DATA 83, 32,205, 79, 67, 75, 69, 78   <008>
6580 DATA 72, 65, 85, 80, 84, 32, 32,203   <248>
6590 DATA 79, 69, 76, 78, 67, 32, 49, 57   <244>
6600 DATA 56, 52, 32,212,225,162,  1,134   <095>
6610 DATA 185,202,134,142,160,208,169,192  <112>
6620 DATA 133,143,169,142, 76,216,255      <136>
6630 :                                     <057>
6640 REM "[SPACE]DIE[SPACE]FOLGENDEN[SPACE]
   KOMMATA[SPACE]SIND                      <086>
6650 REM "[SPACE]'NULLEN',[SPACE]WELCHE[SPACE]
   DIE[SPACE]POKE-                         <242>
6660 REM "[SPACE]SCHLEIFE[SPACE]AUF[SPACE]256
   [SPACE]BYTES[SPACE]ER-                  <214>
6670 REM "[SPACE]GAENZEN.                  <015>
6680 :                                     <108>
6690 DATA,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,  <019>
6700 DATA,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,  <029>
6710 DATA.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,     <243>
```

Listing »Super-Utilities« (Schluß)

Das Fünktionieren Ihrer Füktionstasten hängt fünktionell von den Fünk...

M&T-

H.L. Schneider/W. Eberl

Das Commodore 64-Buch, Bd. 1
1984, 270 Seiten
Der Commodore 64 und seine Handhabung · Einführung in die Grafik · Balkendiagramme · Einführung in die Spritetechnik · Basic-Erweiterungen in Assembler · Ein Leitfaden für Erstanwender.
Best.-Nr. MT 591 (Buch) (Sfr. 44,20/öS 374,40) DM 48,—
Best.-Nr. MT 592 (Beispiele auf Diskette) (Sfr. 58,—/öS 522,—) DM 58,—

H.L. Schneider/W. Eberl

Das Commodore 64-Buch, Bd. 2
1984, 181 Seiten
Spiele nicht nur zum Abtippen · Programmlisting · Programmbeschreibung · Variablenübersicht · Programme nach Anleitung frei ergänzbar · das ideale Buch, um Programmieren spielend zu lernen.
Best.-Nr. MT 593 (Buch) (Sfr. 35,—/öS 296,40) DM 38,—
Best.-Nr. MT 594 (Beispiele auf Diskette) (Sfr. 58,—/öS 522,—) DM 58,—

H. L. Schneider/W. Eberl

Das Commodore 64-Buch, Bd. 3
1984, 206 Seiten
Alles über Sprites · Wissenswertes über Multi-Color-Grafik · Assembler/Disassembler · jede Menge Basic-Erweiterungen · Umgang mit dem Soundgenerator · ein Leitfaden für Fortgeschrittene.
Best.-Nr. MT 595 (Buch) (Sfr. 35,—/öS 296,40) DM 38,—
Best.-Nr. MT 596 (Beispiele auf Diskette) (Sfr. 58,—/öS 522,—) DM 58,—

H. L. Schneider/W. Eberl

Das Commodore 64-Buch, Bd. 4
1984, 261 Seiten
Einführung in Maschinenprogrammierung · Verknüpfung von Maschinenprogrammen mit Basic-Programmen · alles über Assembler/Disassembler · der Leitfaden für Systemprogrammierer
Best.-Nr. MT 597 (Buch) (Sfr. 35,—/öS 296,40) DM 38,—
Best.-Nr. MT 598 (Beispiele auf Diskette) (Sfr. 58,—/öS 522,—) DM 58,—

H. L. Schneider/W. Eberl

Das Commodore 64-Buch, Bd. 5
Juli 1984, 322 Seiten
Ein Leitfaden durch Simon's Basic · ausführliche Besprechung aller Befehle · viele erklärende Beispiele · mit kommentierter Assembler-Listing · das richtige Nachschlagewerk für den geübten Commodore 64-Benutzer.
Best.-Nr. MT 599 (Buch) (Sfr. 35,—/öS 296,40) DM 38,—
Best.-Nr. MT 600 (Beispiele auf Diskette) (Sfr. 58,—/öS 522,—) DM 58,—.

H. L. Schneider/ W. Eberl

Das Commodore 64-Buch, Bd. 6
1984, 190 Seiten
Programmieren auf dem Commodore 64 spielend gelernt · Programmlisting mit anschließender Programmbeschreibung · Variablenübersicht · Tips zum Ändern und Ergänzen des Programms.
Best.-Nr. MT 619 (Buch) (Sfr. 35,—/öS 296,40) DM 38,—
Best.-Nr. MT 620 (Beispiele auf Diskette) (Sfr. 58,—/öS 522,—) DM 58,—

H. L. Schneider

Das Commodore 64-Buch, Bd. 7
August 1984, 110 Seiten
Der Commodore 64 als Klaviatur · Noten schreiben mit hochauflösender Grafik · relative Dateien am Beispiel einer kleinen Adreßverwaltung · Benutzung des Joysticks und der Paddles · Grafikspeicher unter Kernal · Interrupt-Manager · eigene Zeichen definieren · Bildschirmrollen · für Profis.
Best.-Nr. MT 731 (Sfr. 35,—/öS 296,40) DM 38,—

Computerspiele & Wissenswertes — Commodore 64
1984, 156 Seiten
Eine Sammlung von interessanten und nützlichen Maschinenprogrammen · schnelle binäre Artithmetik · Basic-Erweiterungen · mit unterstützendem Assembler-Listing · für den fortgeschrittenen Programmierer.
Best.-Nr. MT 601 (Buch) (Sfr. 27,50/öS 232,40) DM 29,80
Best.-Nr. MT 602 (Beispiele auf Diskette) (Sfr. 38,—/öS 342,—) DM 38,—

F. Ende

Das große Spielebuch — Commodore 64
1984, 141 Seiten
46 Spielprogramme · Wissenswertes über Programmiertechnik · praxisnahe Hinweise zur Grafikherstellung · alles über Joystick- und Paddleansteuerung · das Spielebuch mit Lerneffekt.
Best.-Nr. MT 603 (Buch) (Sfr. 27,50/öS 232,40) DM 29,80
Best.-Nr. MT 604 (Beispiele auf Diskette) (Sfr. 38,—/öS 342,—) DM 38,—

T. Rugg/Ph. Feldman

Mehr als 32 Basic-Programme für den Commodore 64
1984, 279 Seiten
Programme speziell für den Commodore 64 · umfassende praktische Anwendungen · jede Menge Lehr- und Lernhilfen · super Spiele · für Basic-Neulinge und Experten.
Best.-Nr. MT 613 (Buch) (Sfr. 45,10/öS 382,20) DM 49,—
Best.-Nr. MT 614 (Beispiele auf Diskette) (Sfr. 48,—/öS 432,—) DM 48,—

In guten Buchhandlungen, Computershops und Fachabteilungen der Kaufhäuser.

Markt&Technik
Verlag Aktiengesellschaft,
Hans-Pinsel-Str. 2, 8013 Haar bei München, ☎ 089/4613-220
Schweiz: M&T-Vertriebs AG, Alpenstr. 14, CH-6300 Zug, ☎ 042/223155
Österreich: Rudolf-Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, ☎ 0222/677526

-Buchverlag

Dr. P. Albrecht
Commodore 64 — Multiplan
1984, 230 Seiten
Multiplan jetzt auch für den Commodore 64 · der volle Leistungsumfang der 16-Bit-Version · Einführung in die Arbeitsweise von Tabellenkalkulationsprogrammen · praxisnahe Beispiele · Beschreibung alle Befehle und Funktionen · nicht nur für Anfänger.
Best.-Nr. MT 655
(Sfr. 44,20/öS 374,40) **DM 48,—**

E. H. Carlson
Basic mit dem Commodore 64
1984, 320 Seiten
Ein Basic-Lehrbuch für den jugendlichen Anfänger · übersichtlich gegliederte Lernprogramme · Alles über INPUT-GOTO · Let-Befehle · Editorfunktionen · POKE-Befehle für die Grafik · geeignet auch als Leitfaden für Lehrer und Eltern.
Best.-Nr. MT 657
(Sfr. 44,20/öS 374,40) **DM 48,—**

R. E. Williams
CalcResult richtig eingesetzt
1984, 236 Seiten
Ein Übungsbuch speziell für Anwender des CalcResult-Computerprogramms · zahlreiche Einsatzmöglichkeiten im täglichen Leben · Kreditrückzahlung · Rabattberechnung · Kostendeckung · Inventur · Finanzierung und Ankauf eines Hauses und vieles andere mehr.
Best.-Nr. MT 671
(Sfr. 44,20/öS 374,40) **DM 48,—**

W. B. Sanders
Einführungskurs: Commodore 64
1984, 276 Seiten
Die Programmiersprache Basic · Einsatzgebiete des Commodore 64-Basic: Grafik, Musik, Dateiverwaltung · mit vielen Beispielprogrammen, häufig benötigten Tabellen und nützlichen Tips · für Einsteiger und Fortgeschrittene.
Best.-Nr. MT 685
(Sfr. 35,—/öS 296,40) **DM 38,—**

J.W. Willis/D. Willis
Commodore 64 — leicht verständlich
1984, 154 Seiten
Informationen für den Computer-Neuling · Installation und Inbetriebnahme · Programmieren in Basic · Grafik und Töne · Auswahl von Hardware und Zubehör · Software für Ihren Computer · die ideale Einführung in das Arbeiten mit Ihrem Commodore 64.
Best.-Nr. MT 700
(Sfr. 27,50/öS 232,40) **DM 29,80**

G. Beekman
Ihr Heimcomputer Commodore 64
August 1984, 296 Seiten
Alles Wissenswerte im Umgang mit dem Commodore 64 · Planung, Kauf und Inbetriebnahme der Anlage · Einsatz fertig gekaufter oder selbst erstellter Programme · Schwächen und Stärken der altbewährten und neuesten Programmiersprachen · die gängigsten Software-Angebote für jeden Einsteiger ·
Best.-Nr. MT 701
(Sfr. 35,—/öS 296,40) **DM 38,—**

M. J. Winter
Das Commodore 64-LOGO-Arbeitsbuch
September 1984, 225 Seiten
Kinder lernen auf dem Commodore 64 mit der Schildkröte als Lehrer: Bilder malen · Grafikeffekte erzeugen · Wörter verarbeiten · Prozeduren und Variablen · Umgang mit Begriffen wie: Längenmaß, Winkel, Dreieck, Quadrat.
Best.-Nr. MT 720
(Sfr. 31,30/öS 265,20) **DM 34,—**

M. Michalik
35 ausgesuchte Spiele für Ihren Commodore 64
September 1984, 141 Seiten
35 spannende Spiele zum Selbstprogrammieren · Stromschnellen · Nachthimmel · Weltraummanöver · Raketenabwehr · Elektrische Mauer · Heckschütze · Le Maus · Codeknacker · Würfelrallye · Gedankenleser · Schießbude · Dolmetscher u.v.a.m.
Best.-Nr. MT 774
(Sfr. 23,—/öS 193,40) **DM 24,80**

S. Urute
Grafik & Musik auf dem Commodore 64
Oktober 1984, 336 Seiten
68 gut strukturierte und kommentierte Beispielprogramme zur Erzeugung von Sprites und Klangeffekten · Sprite-Tricks · Zeichengrafik · hochauflösende Grafik · Musik nach Noten · spezielle Klangeffekte · Ton und Grafik · für fortgeschrittene Anfänger, die alle Möglichkeiten des C64 ausnutzen wollen.
Best.-Nr. MT 743
(Sfr. 35,—/öS 296,40) **DM 38,—**

M.J.Winter
Lehrspielzeug Computer: C 64/VC-20
Juli 1984, ca. 120 Seiten
Speziell für Kinder entwickelt führt dieses Buch spielerisch in die Basic-Welt des C 64/VC-20 ein · mit vielen lehrreichen Spielprogrammen und Grafikmöglichkeiten · kleinere Kinder benötigen die Hilfe ihrer sachkundigen Eltern.
Best.-Nr. MT 695
(Sfr. 23,—/öS 193,40) **DM 24,80**

In guten Buchhandlungen, Computershops und Fachabteilungen der Kaufhäuser.

Markt&Technik
Verlag Aktiengesellschaft,
Hans-Pinsel-Str. 2, 8013 Haar bei München, ☎ 089/4613-220
Schweiz: M&T-Vertriebs AG, Alpenstr. 14, CH-6300 Zug, ☎ 042/223155
Österreich: Rudolf-Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, ☎ 0222/677526

»Windows« für den C 64

Dieses Programm für den C 64 stellt vier Speicher für Bildschirminhalte im Text/Blockgrafikmodus zur Verfügung. Da diese Speicher im RAM »unterhalb« des Betriebssystems liegen, beanspruchen sie keinen zusätzlichen Speicherplatz. Das Programm ist in Maschinensprache geschrieben und durch Verwendung eines Verschiebeladers in verschiedenen Speicherbereichen lauffähig.

Besonders in größeren Programmen ist es oft wünschenswert, zwischen mehreren Bildschirmen umschalten zu können oder aber Fenster mit Zusatzinformationen in den laufenden Bildschirm einblenden zu können, ohne jedoch seinen ursprünglichen Inhalt zu zerstören. Um dies zu ermöglichen, stellt das Programm vier Speicher zur Verfügung, in denen komplette Bildschirminhalte oder Teile daraus abgelegt werden können. Ein Speicher belegt einen Bereich von 2 KByte, jeweils ein KByte für Zeichen und Farbe. Da diese Speicher unterhalb des Betriebssystems im RAM ab Adresse $E000 untergebracht sind, beanspruchen Sie keinerlei zusätzlichen Platz.

Anwendung

Das Programm bietet folgende Betriebsarten:

— STORE:	SYS BA,NR,XO,YO,FB,FH	Speichern
— RECALL:	SYS BA+3,NR	Rückholen
— SWAP:	SYS BA+6,NR	Austauschen

Die Parameter haben folgende Bedeutung:

BA	=	Basisadresse des Programms (siehe unten)
NR	=	Nummer des angesprochenen Speichers (0...3)
XO,YO	=	Position der linken oberen Ecke des zu speichernden Bereiches (0..39,0..24)
FB	=	Breite des Bereiches (1...40)
FH	=	Höhe des Bereiches (1...25)

Die Fensterseite (Höhe) darf nur bis zum rechten (unteren) Bildrand reichen. Wenn zum Beispiel die linke obere Ecke auf die Koordinate X = 10 und Y = 5 gesetzt ist, so ist die maximale Fensterbreite 30 und die maximale Fensterhöhe 20. Falsche Angaben für die Parameter erzeugen die Fehlermeldung »ILLEGAL QUANTITY ERROR«.

STORE: Um einen kompletten Bildschirm abzuspeichern, geben Sie folgendes ein: SYS BA,NR,0,0,40,25. Mit anderen Parametern erzeugen sie ein Bildschirmfenster, das bei RECALL auch nur den durch diese Parameter definierten Bereich überschreibt. Darum bleibt der ursprüngliche Bildschirm erhalten.

RECALL: Es ist nur die Angabe der Speichernummer nötig. Der gespeicherte Bereich wird an seine ursprüngliche Position zurückgeschrieben.

SWAP: Wenn Sie ein schon vorher vorbereitetes und im Speicher stehendes Informationsfenster in den aktuellen Bildschirm einblenden, wird dessen ursprünglicher Inhalt zerstört. Um ihn zu erhalten, müßten Sie ihn vor der Einblendung in einen noch freien Speicher retten. SWAP erleichtert die Arbeit und spart Speicherplatz, indem es die Inhalte von aktuellem Schirm und Speicher austauscht. Mit einmal SWAP rufen sie den Speicher ab, noch einmal SWAP erzeugt wieder das ursprüngliche Bild. Die Information über Position und Größe des auszutauschenden Bereiches liefert das Bild im Speicher. SWAP wird daher nur mit der Speichernummer aufgerufen.

Noch ein Hinweis: Vermeiden Sie es, mit RECALL oder SWAP einen Speicher aufzurufen, in den noch nichts hineingeschrieben wurde, dies würde zum Absturz des Programms führen (siehe: Kompatibilität zu Simons-Basic). Der Basic-Lader initialisiert alle 4 Speicher, um einer Fehlbedienung vorzubeugen.

Laden des Programmes

Die Verschieberoutine des Basic-Laders gestattet es, das Programm auf den Anfang jeder (sinnvollen) Speicherseite zu legen. Nach dem Start bietet der Lader die Optionen:

1	RAM-Ende normales Basic	($9E00)
2	RAM-Ende Simons-Basic	($7E00)
3	Geschützter Bereich	($C000)
>7	Eingabe der Startseite	

Die Optionen 1 bis 3 setzen den Programmanfang auf die angegebene Adresse, bei Eingabe einer Zahl >7 wird diese als die Nummer derjenigen Speicherseite interpretiert, wo das Programm abgelegt werden soll.

Aus der Nummer der Speicherseite erhält man durch Multiplikation mit 256 die noch fehlende Basisadresse. Das Ladeprogramm gibt sie am Schluß mit aus.

Der Vektor für die Basic-Obergrenze in (55),(56) wird entsprechend der Startadresse des Maschinenprogramms heruntergesetzt, sofern das Programm nicht im geschützten Bereich untergebracht ist. Beachten Sie, daß Sie diesen Vektor nach einem Reset beziehungsweise STOP/RESTORE korrigieren, sonst wird das Programm überschrieben.

Im Programmlisting erkennen Sie zwei Unterschiedlich umfangreiche Blocks von DATA-Zeilen. Der erste, große Block stellt das eigentliche Programm dar, der kleine ist eine Liste von Adressen, welche relativ zum Programmanfang diejenigen Bytes angeben, die bei einer Verschiebung korrigiert werden müssen. Der Korrekturwert ergibt sich aus der Differenz zwischen der Startadresse, mit der das Programm assembliert wurde ($9E00=Speicherseite 158) und der gewünschten Startseite. Die beiden Blocks besitzen eigene Checksummen und OK-/Fehlermeldungen.

Kompatibilität zu Simons-Basic

»Bildspeicher« ist auch eine hilfreiche Erweiterung der Simons-Basic-Befehle für die Bildschirmsteuerung. Es sind jedoch folgende Punkte zu beachten:

— Die Hires-Grafik überschreibt alle vier Speicher. Diese müssen daher vor einem Zugriff mit »RECALL« oder »SWAP« unbedingt neu beschrieben werden.

— Der geschützte Bereich ab $C000 wird ebenfalls von Hires überschrieben, so daß »Bildspeicher« in Programmen, welche die Hires-Grafik benützen, am Basic-RAM-Ende untergebracht werden muß (Option 2).

Ob Sie jedoch nun mit oder ohne Simons-Basic arbeiten, in jedem Falle ist »Bildspeicher« eine nützliche Routine, die auch dem C 64-Besitzer das Fenster zum »Windowing« aufstößt.

(Hans-Herbert Hagedorn / ev)

Listing «Bildspeicher«

```
100 REM    **********************          <143>
110 REM    *                    *          <081>
120 REM    *   BILDSPEICHER     *          <201>
130 REM    *                    *          <101>
140 REM    *   H.H.HAGEDORN     *          <163>
150 REM    *                    *          <121>
160 REM    *   RUPPRECHTSTR.30  *          <202>
170 REM    *                    *          <141>
180 REM    *   83 LANDSHUT      *          <101>
190 REM    *                    *          <161>
200 REM    *   TEL. 0871/67337  *          <199>
210 REM    *                    *          <181>
220 REM    **********************          <007>
230 :                                      <032>
240 PRINT"[CLEAR]":PRINT:PRINT             <006>
250 PRINT"[SPACE5,RVSON]BILDSPEICHER-POSITION
   [SPACE]"                                <249>
260 PRINT:PRINT"[SPACE3]NORMAL-BASIC[SPACE]=
   [SPACE]$9E00[SPACE3](1)"                <206>
265 PRINT:PRINT"[SPACE3]SIMONS-BASIC[SPACE]=
   [SPACE]$7E00[SPACE3](2)"                <226>
270 PRINT:PRINT"[SPACE3]GESCH.BEREICH=[SPACE]
   $C000[SPACE3](3)"                       <001>
275 PRINT:PRINT"[SPACE3]PAGE-ADRESSE[SPACE]
   EINGEBEN[SPACE](>7)"                    <024>
280 PRINT:PRINT TAB(25):INPUT L            <195>
290 IF L=1 THEN L=158                      <032>
300 IF L=2 THEN L=126                      <038>
305 IF L=3 THEN L=192                      <047>
310 IF L>7 AND L<255 THEN D=158-L:GOTO 320 <202>
315 PRINT"???":GOTO 280                    <051>
320 POKE 2,D+100:POKE 780,L:IF L<161 THEN POKE
   56,L:POKE 55,0                          <177>
325 CLR:I=PEEK(780)*256:J=I                <145>
330 READ A:IF A>=0 THEN POKE I,A:I=I+1:S=S+A
   :GOTO 330                               <207>
340 IF S<>51530 THEN PRINT"DATENFEHLER",S:END
                    <116>
345 PRINT"PROGRAMMDATEN[SPACE]OK":S=0
   :D=PEEK(2)-100:IF D=0 THEN 380          <239>
350 READ P:IF P>=0 THEN A=J+P:POKE A,PEEK(A)-D
   :S=S+P:GOTO 350                         <036>
360 IF S<>8737 THEN PRINT"BLOCKKORREKTURFEHLER",
   S:END                                   <043>
370 PRINT:PRINT"BLOCKKORREKTUR[SPACE]OK"   <241>
380 PRINT:PRINT"BASISADRESSE[SPACE]=[SPACE]";J
                    <104>
385 FOR I=0 TO 3:SYS J,I,0,0,40,25:NEXT    <051>
390 DATA 076,009,158,076,087,158,076,091   <035>
400 DATA 158,032,110,158,224,004,176,066   <017>
410 DATA 142,163,159,032,110,158,224,040   <019>
420 DATA 176,056,142,159,159,032,110,158   <047>
430 DATA 224,025,176,046,142,160,159,032   <047>
440 DATA 110,158,224,000,240,036,024,138   <041>
450 DATA 109,159,159,201,041,176,027,202   <069>
460 DATA 142,161,159,032,110,158,224,000   <063>
470 DATA 240,016,024,138,109,160,159,201   <080>
480 DATA 026,176,007,142,162,159,032,116   <097>
490 DATA 158,096,162,014,076,058,164,169   <130>
500 DATA 000,240,002,169,255,141,166,159   <114>
510 DATA 032,110,158,224,004,176,235,142   <118>
520 DATA 163,159,032,197,158,096,032,253   <156>
530 DATA 174,076,158,183,032,005,159,032   <156>
540 DATA 076,159,160,003,185,159,159,145   <177>
550 DATA 091,136,016,248,024,165,091,105   <170>
560 DATA 004,133,091,174,162,159,172,161   <180>
570 DATA 159,177,087,145,091,177,089,145   <222>
580 DATA 093,136,016,245,032,112,159,202   <193>
590 DATA 208,236,096,172,161,159,177,087   <234>
600 DATA 072,177,091,145,087,104,145,091   <229>
610 DATA 177,089,072,177,093,145,089,104   <255>
620 DATA 145,093,136,016,233,032,112,159   <236>
630 DATA 202,208,224,240,059,120,165,001   <233>
640 DATA 072,169,053,133,001,032,076,159   <003>
650 DATA 160,003,177,091,153,159,159,136   <023>
660 DATA 016,248,032,005,159,024,169,004   <021>
670 DATA 101,091,133,091,174,162,159,173   <035>
680 DATA 166,159,208,183,172,161,159,177   <067>
690 DATA 091,145,087,177,093,145,089,136   <078>
700 DATA 016,245,032,112,159,202,208,236   <056>
710 DATA 104,133,001,088,096,032,067,159   <076>
720 DATA 024,172,160,159,240,016,173,164   <082>
730 DATA 159,105,040,141,164,159,144,003   <088>
740 DATA 238,165,159,136,208,240,024,173   <112>
750 DATA 164,159,109,159,159,141,164,159   <137>
760 DATA 133,087,133,089,144,003,238,165   <132>
770 DATA 159,024,173,136,002,109,165,159   <141>
780 DATA 133,088,024,169,216,109,165,159   <161>
790 DATA 133,090,096,169,000,141,164,159   <159>
800 DATA 141,165,159,096,169,000,133,091   <139>
810 DATA 133,093,169,224,133,092,169,228   <189>
820 DATA 133,094,174,163,159,240,016,024   <186>
830 DATA 160,008,152,101,092,133,092,152   <182>
840 DATA 101,094,133,094,202,208,243,096   <203>
850 DATA 024,169,040,101,087,133,087,144   <214>
860 DATA 002,230,088,024,169,040,101,089   <218>
870 DATA 133,089,144,002,230,090,056,173   <231>
880 DATA 161,159,101,091,133,091,144,002   <233>
890 DATA 230,092,056,173,161,159,101,093   <255>
900 DATA 133,093,144,002,230,094,096,000   <253>
905 DATA -1                                <109>
910 REM KORREKTURTABELLE                   <226>
920 :                                      <213>
930 DATA 002,005,008,011,018,021           <129>
940 DATA 028,031,038,041,050,058           <158>
950 DATA 061,070,077,080,095,098           <187>
960 DATA 105,108,118,121,126,141           <174>
970 DATA 144,158,165,191,207,214           <202>
980 DATA 220,230,233,238,252,263           <200>
990 DATA 267,272,277,282,289,292           <246>
1000 DATA 295,304,311,319,327,330          <229>
1010 DATA 348,393,405,-1                   <049>
```

DATA-Erzeuger

Dieses Programm dient, wie der Name schon sagt, zum automatischen Erzeugen von DATA-Zeilen. Da das Programm in Maschinensprache geschrieben ist, ist die Ausführungszeit entsprechend kurz. Der Inhalt von 10000 Speicherzellen wird in weniger als 4 Sekunden in DATA-Zeilen umgewandelt!

Es kommt immer wieder vor, daß der Inhalt von Speicherzellen in DATA-Zeilen umgewandelt werden muß. Verschiedene Lösungen habe ich aus irgendwelchen Büchern oder Zeitschriften abgetippt. Da ich keine dieser Programme als befriedigend empfand, begann ich, einen DATA-Erzeuger in Basic zu programmieren. Auch dieses Programm war zu unflexibel und vor allem zu langsam. Da ich zwischenzeitlich verstärkt in Maschinensprache programmiere, bestand ein noch größerer Bedarf an einem leistungsfähigen DATA-Erzeuger. Also beschloß ich, einen solchen in Maschinensprache zu erstellen.

Leicht zu bedienen

Durch eine umfangreiche Fehlerprüfung ist das Programm sehr anwendungssicher. Die DATA-Zeilen können, falls gewünscht, an ein Basic-Programm angehängt werden, und das beliebig oft.

Eingabe-Format:

SYS 49152,ANFADR,ENDADR,ANFZNR,SCHRITTWEITE

ANFADR:	Adresse der ersten Speicherzelle, aus deren Inhalt DATA-Zeilen erzeugt werden sollen.
ENDADR:	Adresse der letzten Speicherzelle für die DATA-Zeilen-Erzeugung.
ANFZNR:	Zeilennummer der ersten DATA-Zeile (muß größer sein als die letzte Zeilennummer im Basic-Programm, falls eines im Speicher ist).
SCHRITTWEITE:	Um diese Zahl wird die Zeilennummer bei der DATA-Erzeugung jeweils erhöht.

Fehlermeldungen

Für ANFADR, ENDADR, ANFZNR dürfen nur ganze, positive Zahlen von 0-65536 eingegeben werden. Die SCHRITTWEITE darf zwischen 1 und 255 liegen. Werden Zahlen eingegeben, die außerhalb des oben genannten Bereichs liegen, so kommt es zur Fehlermeldung »? ILLEGAL QUANTITY ERROR«. Weitere Fehlermeldungen sind:

- ANFANGSADRESSE GROESSER ALS ENDADRESSE.
  FEHLER! Logischerweise muß die Anfangsadresse kleiner sein als die Endadresse.
- 0 ALS SCHRITTWEITE NICHT MOEGLICH!
  Würde 0 als Schrittweite akzeptiert werden, so hätten alle erzeugten DATA-Zeilen die gleiche Zeilennummer.
- ZEILENNUMMER GROESSER ALS 63999!
  Bei der Erzeugung der DATA-Zeilen wurde eine Zeilennummer erzeugt, die größer als 63999 war. Da der C 64 nur Zeilennummern bis 63999 zuläßt, würden größere Zeilennummern Probleme der Art aufwerfen, daß sie sich nicht mehr löschen oder ändern ließen. Man sollte eine kleinere Anfangs-Zeilennummer oder Schrittweite wählen, um den Fehler zu beseitigen.
- FREIER SPEICHERPLATZ REICHT NICHT!
  Bei der DATA-Erzeugung wurde festgestellt, daß der freie RAM-Bereich nicht ausreicht.

Beim Auftreten eines Fehlers wird die DATA-Erzeugung, falls schon begonnen, sofort abgebrochen. Die bis dahin erzeugten DATA-Zeilen werden wieder gelöscht. Die Programmzeiger werden wieder so eingestellt, wie es vor dem Aufruf des DATA-Erzeugers der Fall war.

War die DATA-Zeilen-Erzeugung erfolgreich, so erscheint die Meldung »DATA ERZEUGER FERTIG! (C) J. MATERNA«. Jetzt befinden sich die DATA-Zeilen und die Schleife zum Lesen und ZurückPOKE der DATAs im Speicher. Das Basic-Programm kann wie gewohnt gelistet, gestartet oder gespeichert werden.

Wenn man möchte, kann man noch beliebige weitere DATA-Zeilen erzeugen. Sie werden automatisch an die bereits vorhandenen angehängt. Wichtig: Die neue Anfangszeilennummer muß größer sein als die letzte im Programm verwendete. Am besten LISTet man das Programm erst auf, um die höchste Zeilennummer festzustellen. Möchte man die DATA-Blöcke als Unterprogramme aufrufen, muß natürlich noch am Ende des DATA-Blocks eine Zeilennummer mit »RETURN« eingegeben werden.

Tips und Beispiele

Bei allen Bereichen wird vorausgesetzt, daß sich der DATA-Erzeuger lauffähig im Speicher befindet, ebenso die Daten, aus denen DATA-Zeilen erzeugt werden sollen.

Sprite-Blöcke übernehmen

- Das Programm, das die Sprites erzeugt, laden und starten. Wenn die Sprites erzeugt sind, das Programm abbrechen und die Adressen der Sprites aufschreiben.
- Das Programm mit NEW löschen.
- Das eigene Programm, das die Sprites verwenden soll, laden und die höchste Zeilennummer feststellen.
- Beispielsweise SYS 49152, Anfangsadresse der Sprites, Endadresse der Sprites, höchste Zeilennummer des eigenen Programms plus 10, sowie beliebige Schrittweite eingeben.
- Den letzten Punkt beliebig oft wiederholen, bis alle Sprites in DATA-Zeilen umgewandelt sind. Dabei darf nicht vergessen werden, stets erneut die höchste Zeilennummer festzustellen.
- Es ist sinnvoll, am Ende der DATA-Blöcke jeweils eine Zeile mit »RETURN« einzugeben. Danach können die Sprites irgendwo vom Hauptprogramm aus mit einem GOSUB-Befehl aufgerufen werden.
- Programm testen und speichern.

Nach dem oben beschriebenen Verfahren kann man natürlich nicht nur Sprites übernehmen, sondern auch Maschinenprogramme und neue Zeichensätze. Insbesondere kann man auch sehr gut Maschinenprogramme, die von einem Assembler erzeugt wurden, an ein Basic-Programm anhängen.

Maschinenprogramm für Datenfernübertragung vorbereiten

Da die Daten im ASCII-Format übertragen werden, müssen sie vorher umgewandelt werden.

Beispiel: Das zu übertragende Programm ist 8000 Bytes lang und wird normal ab Basic-Anfang geladen.

- Das zu übertragende Programm laden.
- POKE 44,40 : POKE 40 * 256,0 : NEW
  Damit ist das Programm vor dem Überschreiben geschützt. Wenn das Maschinenprogramm am Speicherende liegt, dann muß natürlich der Zeiger für das Basic-Ende herabgesetzt werden.
- SYS 49152,2049,10049,100,2

Nun muß das Programm nur noch als Datei gespeichert werden und kann dann von einem Terminal-Programm gesendet werden.

Ein Monitor-Programm ($9000 — $9FFF) in DATA-Zeilen verwandeln.

- Monitor laden.
- SYS 49152,36864,40959,100,2.
- Abspeichern
  Das Programm kann nun normal geladen und gestartet werden. Danach kann der Monitor mit der entsprechenden SYS-Adresse aufgerufen werden.

Bildschirminhalt in DATA-Zeilen verwandeln

SYS 49152,1024,2023,100,2

Das Programm kann sofort gestartet werden.

Programmierte DATA-Erzeugung

Der DATA-Erzeuger läßt sich auch von Basic-Programmen aus aufrufen. Dabei dürfen die Parameter natürlich in Variablen übergeben werden. Doch Achtung: Wenn der DATA-Erzeuger fertig ist, wird ein CLR ausgeführt und zum Basic-Warmstart gesprungen. Dabei wird das Basic-Programm verlassen.

Um wieder ins Basic-Programm zurückzukommen, empfehle ich folgenden Trick, den ich am besten durch ein Beispiel demonstriere:

```
10 A=900:Z=100
20 A=A+100:Z=Z+20:IF Z=200 THEN END
30 PRINT CHR$(147);:PRINT CHR$(17):PRINT CHR$(17)
40 PRINT"A=";A;":Z=";Z;":GOTO 20";CHR$(19)
50 POKE 631,13:POKE 198,1
60 SYS 49152,A,A+100,Z,2:END
```

Dieses Programm hält in Zeile 60 an und startet sich so lange wieder automatisch, bis in Zeile 20 Z=200 ist. Dabei werden die Variablen, die in Zeile 40 auf dem Bildschirm ausgegeben werden, mit übernommen. Für den Auto-Start des Programms ist die Zeile 50 verantwortlich. Dort wird ein RETURN (POKE 631,13) in den Tastaturpuffer gePOKEt und die Länge des Tastaturpuffers (POKE 198,1) auf 1 gesetzt.

Anhand dieses Beispiels sollte jeder in der Lage sein, dieses Problem zu lösen.

Ich hoffe, daß ich durch meine Beispiele genug Anregungen für den Einsatz des DATA-Erzeugers gegeben habe.

Die Programmeingabe

Nachdem das Programm in den C64 eingegeben ist, kann es bedenkenlos gestartet werden.

Falls dann die Meldung «DATA-FEHLER« erscheint, müssen die DATA-Zeilen noch einmal überprüft werden. Wenn die Meldung «PROGRAMM OK« erscheint, speichert man das Programm am besten ab, bevor man den DATA-Erzeuger aufruft. Vor dem Programmstart mit »SYS« löscht man am besten den Speicher mit »NEW« oder lädt das Programm, für das DATAs erzeugt werden sollen.

Möchte man den DATA-Erzeuger als Maschinenprogramm direkt speichern, so sind folgende Befehle einzugeben:

POKE 43,0:POKE 44,192:POKE 45,66:POKE 46,195:SAVE "MP-DATA-ERZEUGER",8,1

Für die Datasette muß statt der 8 natürlich eine 1 eingegeben werden.

Das so gespeicherte Programm lädt man mit »LOAD "MP-DATA-ERZEUGER",8,1«.

Nachdem man NEW eingegeben hat, ist das Programm sofort einsatzbereit.

(Jörg Materna / ev)

```
10 REM **********************************  <217>
15 REM *                                *  <242>
20 REM *           DATA-MAKER           *  <174>
30 REM *                                *  <001>
40 REM *         JOERG MATERNA          *  <138>
50 REM *         BECKUMER STR.91        *  <244>
60 REM *         4730 AHLEN             *  <085>
65 REM *                                *  <036>
70 REM **********************************  <021>
75 REM                                     <218>
80 PRINT CHR$(147)                         <157>
100 FOR A=49152 TO 49986:READ X:POKE A,X:SU=SU+X
   :NEXT                                   <062>
105 IF SU<>102090 THEN PRINT "DATA-FEHLER" <173>
106 PRINT "PROGRAMM[SPACE]OK"              <070>
110 DATA 169,0,141,65,195,32,233,193,132,251,
   133,252,32,233,193,140,54,195,141,55    <212>
120 DATA 195,32,233,193,140,59,195,141,60,195,
   32,253,174,32,158,183,142,63,195,138    <042>
130 DATA 208,3,76,58,194,165,252,205,55,195,144,
   12,208,7,165,251,205,54,195,144         <156>
140 DATA 3,76,68,194,56,165,45,233,2,141,61,195,
   133,253,165,46,233,0,141,62,195         <157>
150 DATA 133,254,160,2,173,59,195,145,253,200,
   173,60,195,145,253,200,169,129,145      <214>
160 DATA 253,200,169,65,145,253,200,169,178,145,
   253,200,152,72,165,251,166,252,32       <014>
170 DATA 243,193,141,64,195,104,168,32,7,194,
   169,164,145,253,200,152,72,173,54,195   <089>
180 DATA 174,55,195,32,243,193,141,64,195,104,
   168,32,7,194,169,58,145,253,200,169     <059>
190 DATA 135,145,253,200,169,88,145,253,200,169,
   58,145,253,200,169,151,145,253,200      <094>
200 DATA 169,65,145,253,200,169,44,145,253,200,
   169,88,145,253,200,169,58,145,253       <025>
210 DATA 200,169,130,145,253,76,52,193,160,2,
   173,59,195,145,253,200,173,60,195,145   <117>
220 DATA 253,200,169,131,145,253,200,152,72,160,
   0,177,251,32,160,193,104,168,173        <005>
230 DATA 56,195,201,48,240,3,145,253,200,173,57,
   195,201,48,208,10,173,56,195,201        <029>
240 DATA 48,240,6,173,57,195,145,253,200,173,58,
   195,145,253,165,251,205,54,195,208      <163>
250 DATA 13,165,252,205,55,195,208,6,238,65,195,
   76,52,193,230,251,208,2,230,252         <009>
260 DATA 192,74,176,8,200,169,44,145,253,76,223,
   192,200,169,0,145,253,200,140,64        <062>
270 DATA 195,160,0,24,165,253,109,64,195,145,
   253,141,64,195,165,254,105,0,200,145    <119>
280 DATA 253,133,254,197,56,208,3,76,45,194,173,
   64,195,133,253,173,65,195,208,28        <119>
290 DATA 24,173,59,195,109,63,195,141,59,195,
   144,13,238,60,195,173,60,195,201,250    <166>
300 DATA 208,3,76,32,194,76,205,192,24,165,253,
   105.,2,133,45,165,254,133,46,144,2      <100>
310 DATA 230,46,160,0,152,145,253,200,145,253,
   32,78,194,32,96,166,76,134,227,162      <114>
320 DATA 0,142,56,195,142,57,195,142,58,195,201,
   100.144,8,233,100,238,56,195,76         <080>
330 DATA 171,193,201,10,144,8,233,10,238,57,195,
   76,183,193,201,0,240,8,238,58,195       <186>
```

```
340 DATA 233,1,76,195,193,24,169,48,109,56,195,
    141,56,195,169,48,109,57,195,141          <139>
350 DATA 57,195,169,48,109,58,195,141,58,195,96,
    32,253,174,32,138,173,32,247,183          <201>
360 DATA 96,133,99,134,98,162,144,56,32,73,188,
    32,223,189,32,135,180,32,166,182          <144>
370 DATA 96,162,0,165,34,141,20,194,165,35,141,
    21,194,189,0,1,145,253,200,232,206        <202>
380 DATA 64,195,208,244,96,32,86,194,169,120,
    160,194,32,30,171,76,154,193,32,86        <161>
390 DATA 194,169,155,160,194,32,30,171,76,154,
    193,169,192,160,194,32,30,171,76,154      <058>
400 DATA 193,169,229,160,194,32,30,171,76,154,
    193,169,11,160,195,32,30,171,96,173       <015>
410 DATA 61,195,133,253,173,62,195,133,254,160,
    0,152,145,253,200,145,253,24,173          <155>
420 DATA 61,195,105,2,133,45,173,62,195,105,0,
    133,46,96,18,90,69,73,76,69,78,78         <207>
430 DATA 85,77,77,69,82,32,71,82,79,69,83,83,69,
    82,32,65,76,83,32,54,51,57,57,57          <045>
440 DATA 33,146,0,18,70,82,69,73,69,82,32,83,80,
    69,73,67,72,69,82,80,76,65,84,90          <034>
450 DATA 32,82,69,73,67,72,84,32,78,73,67,72,84,
    33,146,0,18,48,32,65,76,83,32,83          <034>
460 DATA 67,72,82,73,84,84,87,69,73,84,69,32,78,
    73,67,72,84,32,77,79,69,71,76,73          <090>
470 DATA 67,72,33,146,0,18,65,78,70,65,78,71,83,
    65,68,82,69,83,83,69,32,62,32,69          <068>
480 DATA 78,68,65,68,82,69,83,83,69,46,70,69,72,
    76,69,82,33,146,0,18,42,32,68,65          <089>
490 DATA 84,65,32,69,82,90,69,85,71,69,82,32,70,
    69,82,84,73,71,33,32,40,67,41,32          <070>
500 DATA 74,46,77,65,84,69,82,78,65,46,32,42,
    146,0,66,195,49,52,50,244,1,1,8,10        <018>
510 DATA 48,0,0                               <166>
```

Single-Step für VC 20

»Single-Step« soll bei Maschinenprogrammen zum Aufdecken von Fehlern beitragen. Anfängern erlaubt es ein schnelleres Einfühlen in die Wirkungsweise der Maschinenbefehle.

Das Programm ist für den Commodore VC 20 mit mindestens 8 KByte Speichererweiterung geschrieben (ein entsprechendes Programm für den C 64 finden Sie im 64'er Stammagazin). Es ist ein DATA-Listing einer Maschinenspracheroutine. Über Zeile 9 des Listings läßt sich die gewünschte Startadresse festlegen; die Basic-Schleife ab Zeile 300 korrigiert absolute Sprungadressen. Hierzu wurden bei der Programmerstellung die absoluten Zieladressen so eingestzt, als stünde das Programm ab Adresse 0 im Speicher. Der Basic-Teil sorgt dafür, daß die Startadresse (ss) als Offset berücksichtigt wird.

Nach einmaligem Starten läßt sich die Maschinenspracheroutine mit SYS ss starten, der Basic-Teil kann demnach gelöscht werden, sollte aber sicherheitshalber vorher abgespeichert werden.

Für die Bedienung des Programmes stehen die Funktionstasten F1, F3 und F5 zur Verfügung. Ein Beispiel soll den Ablauf verdeutlichen (Tabelle):

Zunächst startet man »Single-Step« mit SYS ss, es erscheint dann die Überschrift mit Bezeichnung der einzelnen Bildschirmspalten und der ersten, übersetzten Zeile. Nun kann man über F3 in den Editier-Modus gelangen, es lassen sich sämtliche dargestellte Register und Flags ändern. Dazu fährt man mit dem Cursor einfach an die entsprechende Stelle und tippt die gewünschten Werte ein. Im Editier-Modus wird nun der PC (Programm-Counter) so geändert, daß er auf die Startadresse der eigenen Routine zeigt (hier im Beispiel ist dies die Adresse $ 4000). Durch Betätigen der Taste F1 wird der hier abgelegte Befehl ausgeführt. Der PC zeigt dann auf die Startadresse des nächsten Befehles, PSW (Programm Status Word, auch Flag-Register genannt),A,X und Y sind entsprechend verändert. (Im Beispiel wurde der Akku mit den Daten #00 geladen). Jede weitere Betätigung der Funktionstaste F1 bringt einen weiteren Befehl zur Ausführung.

Jetzt noch eine Anmerkung zu Manipulationen im PSW (Flag-Register): Durch Ändern einzelner Flags lassen sich zum Beispiel Schleifen vorzeitig beenden oder, wie im Beispiel in Zeile 16 geschehen, auch verlängern; das Zero-Bit wurde zurückgesetzt, was die CPU durch einen weiteren relativen Sprung auf Adresse $ 4000 quittieren mußte.

Weiterhin lassen sich unter Benutzung des PSW, genauer des I-Flags, Breakpoints im Programm setzen. Es ist nicht immer nötig das Programm schrittweise von Anfang bis Ende zu untersuchen. In solchen Fällen setzt man zu Beginn des inter-

```
                    ** Beispiel **
$4000       lda #0    a900    ;sum gleich null
$4002       tax       aa      ;index gleich null
$4003       sed       f8      ;dezimale addition
$4004  sum  clc       18      ;übertrag nicht berücksichtigen
$4005       adc $2,x  7502    ;sum = sum+daten
$4007       inx       e8
$4008       cpx $0    e400    ;alle elemente ?
$400a       bne sum   d0f8    ;nein, dann weiter
$400c       sta $1    8501    ;lege Ergebnis ab
$400e       rts       60

$0000       hex       02      ;anzahl elemente
$0001                 00      ;platz für summe reservieren
$0002                 793327  ;zu add. elemente in packed bcd

Ablauf mit Single-Step:

00  :** single   —   step **  :
01  :p c  nv-bdizc  a  x  y   :
02  :                         :
03  :e14e-00-00011-00-00-2c: ;ändern in →
04  :4000-          - ff - ff -  : ;next step führt zu →
05  :4002-00-00011-00- ff -2c:
06  :4003-00-00011-00-00-2c: ;next step = set decimal flag →
07  :4004-00-01011-00-00-2c: ;clear carry →
08  :4005-00-01010-00-00-2c:
09  :4007-00-01000-79-00-2c: ;inkrementiere x-reg. →
10  :4008-00-01000-79-01-2c: ;vergleiche, setze n, z, c →
11  :400a-10-01000-79-01-2c: ;nicht null also sprung →
12  :4004-10-01000-79-01-2c:
13  :4005-10-01000-79-01-2c: ;addiere, setze n,v,z,c →
14  :4007-11-01001-12-01-2c: ;inkrementiere x-reg. →
15  :4008-01-01001-12-02-2c: ;vergleiche, setze n,z,c →
16  :400a-01-01011-12-02-2c: ;null also kein sprung →
17  :400c-01-01011-12-02-2c: ;ergebnis ablegen ...

mit Änderung:

16  :400a-01-01011-12-02-2c: ;ändern →
16' :400a-        0          : ;erzwingt sprung →
17' :4004-01-01001-12-02-2c:
18' :4005-01-01000-12-02-2c: ;addiere →
19' :4007-00-01000-39-02-2c: ;inkrementiere x-reg. →
20' :4008-00-01000-39-03-2c:
```

Beispielgrogramm und Ablauf mit »Single Step«

essierenden Teiles den Befehl CLI ein. Es genügt nun im Editiermodus das I-Flag zu setzen und mit F1 den nächsten Schritt zu starten. Das Programm wird erst wieder nach Abarbeiten des CLI- und des darauffolgenden Befehles angehalten.

Die Taste F5 dient zum Ausstieg aus der »Single-Step«-Routine. Ein Betätigen dieser Taste hat die gleiche Wirkung wie Stop/Restore, das heißt es wird eine teilweise Neuinitialisierung und ein Sprung zum Basic-Warmstart durchgeführt.

Abschließend noch eine Anmerkung zur Wirkungsweise:

Ich benutze hier den Timer 1 des VIA 6522 (versatile interface adapter) in der Betriebsart »One-Shot«. Diese läßt sich über das ACR (auxiliary control register) definieren. Der Timer erzeugt jetzt nach jedem Start einen einzelnen Interrupt. Durch geeignete Wahl des Timer-Zählerstandes kann erreicht werden, daß nach jedem Rücksprung aus der Interruptroutine mittels RTI jeweils ein Befehl des zu untersuchenden Programmes ausgeführt wird und darauffolgend die weitere Ablaufsteuerung wieder von der Interruptroutine übernommen wird. Im ersten Hauptteil der Routine »Single-Step« hole ich dann die interessierenden Daten vom Stack und stelle sie auf dem Bildschirm dar. Vor dem Rücksprung mittels RTI wird dann im zweiten Teil die eventuell geänderte Zeile wieder eingelesen und übersetzt.

(Hermann Weißenberger / ev)

```
0 REM ** HERMANN WEISSENBERGER **                  <021>
1 REM TEL 09722/2672                               <121>
2 :                                                <060>
3 REM ::::::::::::::::::::::::                     <002>
4 REM : ** SINGLE - STEP **  :                     <218>
5 REM :PC   NV-BDIZC A  X  Y :                     <202>
6 REM ::::::::::::::::::::::::                     <005>
7 :                                                <065>
8 REM STARTADRESSE --> SS                          <106>
9 :   SS=24576: REM $6000                          <086>
10 :                                               <068>
11 REM ROUTINE VERSCHIEBBAR UEBER SS               <073>
12 :                                               <070>
13 POKE 56,INT(SS/256+.001)                        <154>
14 POKE 55,INT(SS-256*PEEK(56)+.001)               <194>
15 :                                               <073>
16 REM *********************                       <059>
17 REM DATALISTE DER ROUTINE (438 BYTE)            <096>
18 :                                               <076>
20 FOR I=SS TO SS+437: READ W: POKE I,W: NEXT
                        <057>
22 DATA 160,0,185,49,0,32,210,255,200,192          <233>
24 DATA 44,208,245,120,173,43,145,41,63,141
                        <087>
26 DATA 43,145,169,93,141,20,3,169,0,141           <199>
28 DATA 21,3,169,128,141,138,2,169,20,141          <248>
30 DATA 36,145,169,0,141,37,145,88,96,147          <021>
32 DATA 32,42,42,32,83,73,78,71,76,69              <045>
34 DATA 32,45,32,83,84,69,80,32,42,42              <067>
36 DATA 13,80,67,32,32,32,78,86,45,66              <077>
38 DATA 68,73,90,67,32,65,32,32,88,32              <082>
40 DATA 32,89,13,173,36,145,169,0,72,40            <172>
42 DATA 32,46,1,169,13,32,210,255,160,21           <203>
44 DATA 169,45,32,210,255,136,16,248,186,160
                        <173>
46 DATA 0,32,57,1,32,57,1,200,32,85                <213>
48 DATA 1,200,32,57,1,200,32,57,1,200              <042>
50 DATA 32,57,1,32,159,255,32,228,255,201          <019>
52 DATA 135,240,44,201,134,240,44,201,133,208
                        <199>
54 DATA 238,186,160,0,32,112,1,32,112,1            <156>
56 DATA 200,32,148,1,200,32,112,1,200,32           <191>
58 DATA 112,1,200,32,112,1,169,0,141,37            <154>
60 DATA 145,104,168,104,170,104,64,120,76,210
                        <216>
62 DATA 254,169,145,32,210,255,160,0,177,209
                        <183>
64 DATA 9,128,145,209,152,72,32,46,1,32            <190>
66 DATA 159,255,32,228,255,201,0,240,246,170
                        <182>
68 DATA 104,168,177,209,41,127,145,209,224,133
                        <036>
70 DATA 240,175,224,135,240,207,224,29,240,28
                        <235>
72 DATA 224,157,240,37,224,48,144,206,224,71
                        <193>
74 DATA 176,202,224,58,144,4,224,65,144,194
                        <151>
76 DATA 177,209,41,127,201,45,240,186,138,32
                        <198>
78 DATA 210,255,200,192,22,144,177,169,157,208
                        <045>
80 DATA 5,192,0,240,169,138,32,210,255,136 <096>
82 DATA 16,162,160,120,162,221,202,208,253,136
                        <027>
84 DATA 208,248,96,189,6,1,74,74,74,74           <191>
86 DATA 32,73,1,189,6,1,41,15,202,201            <096>
88 DATA 10,144,2,233,57,105,48,145,209,200 <096>
90 DATA 96,189,6,1,72,104,10,72,169,48           <181>
92 DATA 144,2,169,49,192,7,240,2,145,209         <019>
94 DATA 200,192,13,144,236,104,202,96,177,209
                        <005>
96 DATA 32,139,1,10,10,10,10,157,6,1             <042>
98 DATA 200,177,209,32,139,1,29,6,1,157          <223>
100 DATA 6,1,200,202,96,201,48,176,2,105         <212>
102 DATA 9,41,15,96,189,6,1,41,32,157            <088>
104 DATA 6,1,169,128,72,177,209,201,49,208 <086>
106 DATA 8,104,72,29,6,1,157,6,1,200             <023>
108 DATA 104,74,72,208,236,104,202,96            <089>
200 :                                            <002>
205 REM *********************                    <248>
210 REM RELATIVE PLAETZE DER                     <173>
215 REM ABSOLUTEN SPRUNGADRESSEN                 <071>
220 :                                            <022>
225 DATA 3,101,122,125,129,133,137,141,165 <182>
230 DATA 168,172,176,180,184,217,321,371,384
                        <064>
235 DATA 23,28: REM 'INTERRUPT-VEKTOR'           <085>
240 :                                            <042>
245 REM *********************                    <032>
250 :                                            <052>
300 REM ABSOLUTE SPRUNGADRESSEN KORRIGIEREN
                        <144>
305 :                                            <108>
310 FOR I=0 TO 17                                <239>
315 :   READ A                                   <062>
320 :   X1=PEEK(SS+A)+PEEK(SS+A+1)*256+SS  <028>
325 :   X2=INT(X1/256+.001)                      <254>
330 :   X1=INT(X1-256*X2+.001)                   <054>
335 :   POKE A+SS,X1                             <103>
340 :   POKE A+SS+1,X2                           <072>
345 NEXT I                                       <037>
350 :                                            <153>
355 READ A1:X1=PEEK(SS+A1)                       <167>
360 READ A2:X1=X1+PEEK(A2)*256+SS                <042>
365 X2=INT(X1/256+.001)                          <236>
370 X1=INT(X1-256*X2+.001)                       <036>
375 POKE A1+SS,X1                                <134>
380 POKE A2+SS,X2                                <141>
385 :                                            <188>
390 REM *********************                    <178>
395 :                                            <198>
400 REM BELEGUNG DER FUNKTIONSTASTEN             <212>
405 :                                            <208>
410 REM F1 = NEXT STEP                           <089>
415 REM F3 = EDIT-MODE                           <093>
420 :   REM AKZEPTIERT WERDEN TASTEN 0...F UND
                        <236>
425 :   REM CURSOR-RIGHT,CURSOR-LEFT             <094>
430 REM F5 = AUSSTIEG                            <091>
435 :                                            <238>
500 PRINT                                        <142>
505 PRINT"[SPACE2]START[SPACE]MIT:"              <137>
510 PRINT"SYS";SS                                <188>
515 END                                          <133>
```

Disksorter in Vollendung

Dieses Verwaltungs- und Sortierprogramm für Disketten läßt keine Wünsche offen. Es macht aus Ihrem Diskettendschungel eine übersichtliche Sache. So können Sie sämtliche Einträge entweder alphabetisch oder nach ID ordnen und sortieren. Auf Wunsch werden doppelte Einträge ausgesondert. Wenn Sie Ihre eigene Liste mit einer anderen Liste vergleichen wollen, wird registriert, welche Programme in Ihrer Liste nicht vorhanden sind. Aber das ist noch längst nicht alles.

In diesem Heft ist die Basic-Version abgedruckt. Das bedeutet, daß die Sortierroutine etwas langsam arbeitet. Auf der ebenfalls erhältlichen Diskette zum Heft finden Sie eine compilierte Version, bei der auch die Sortierroutine in Maschinensprache geschrieben ist. Diese Version stellt wohl das optimalste dar. Doch nun zu den Funktionen des Programms.

Anleitung

In den Speicher Ihres Commodore C 64 können zirka 850 Programmdaten gespeichert werden. Ein Datensatz besteht

```
DATEN PER DISK EINLESEN
DATEN ERWEITERN / LOESCHEN
LISTE EDITIEREN
LISTE SORTIEREN
DATEN ABSPEICHERN
DATEN EINLESEN              QUIT
LISTE DRUCKEN               SPEZ.LISTE
LISTE AUF SCREEN            DISK OPER.
BLOCKS FREE                 DIRECTORY
ANDERE BILDSCHIRM FARBE
AUSSORTIEREN FREMDER LISTE
```

Bild 1. Nach dem Starten des Programms wird das Hauptmenü erstellt. Von hier aus sind sämtliche Funktionen anwählbar.

```
: ELEMENT LOESCHEN
: ELEMENT SUCHEN
: ELEMENT VERBESSERN
:
: ELEMENT DRUCKEN
: EIN ELEMENT VOR
: 20 ELEMENTE VOR
: 20 ELEMENTE ZURUECK
: EIN ELEMENT ZURUECK
: DM-LISTE ERSTELLEN
: ELEMENT EINFUEGEN
```

Bild 2. Diese Möglichkeiten haben Sie, wenn vom Hauptmenü (siehe Bild 1) »F5 Liste editieren« gewählt wurde.

aus dem Programmnamen, der ID-Nummer und der Anzahl der belegten Blocks auf Diskette.

Achtung: Alle Eingaben die nicht mit J/N beantwortet werden können, müssen mit RETURN abgeschlossen werden.

F1: Setzt den Datenzähler auf Null, das Erstellen der Liste beginnt. Daten werden von Diskette eingelesen. Sie können wählen, ob Sie alle Daten einlesen wollen, oder einzelne Daten in die Liste von Diskette aufnehmen möchten. Falls Sie mehr als eine Diskette einlesen wollen, die Anfrage des Computers mit »J« beantworten, sonst erfolgt eine Rückkehr in das Hauptmenü. Wollen Sie in diesem Fall zusätzliche Daten aufnehmen, ist die Funktion F3 zu wählen.

F3: F1 liest Daten zusätzlich zu vorhandenen Daten ein. Mit F3 kann ein schon vorhandenes Datenfile von Diskette zugeladen werden.

Mit F5 können alle Daten einer ID-Nummer von einer in den Computer eingelesenen Directory gelöscht werden.

Mit F7 kann der gerade im Computer befindliche Standard-Druckkopf abgespeichert werden (automatisch unter Standardkopf). Der Standardkopf wird sofort nach dem Laden des Programms eingelesen und dient dazu, Name, Adresse, Telefonnummer (oder andere Daten) in Sperrschrift über die Liste zu drucken.

F5 erlaubt das Editieren von Daten. Daten können gelöscht, verändert, zusätzlich eingefügt und direkt auf den Drucker ausgegeben werden. Zusätzlich besteht die Möglichkeit in diesem Modus, aus einer Programmliste eine Preisliste zu machen.

F7 sortiert alle Daten alphabetisch oder nach ID-Nummer. Danach können doppelte Daten aussortiert werden, Eingabe der Anzahl zu berücksichtigender Zeichen von links.

F2 speichert die Daten unter dem eingegebenen Filenamen ab. Klammeraffe kann verwendet werden.

F4 liest ein gespeichertes Datenfile in den Rechner ein, der Joker ist erlaubt.

F6 gibt die Daten auf den Drucker aus. Entweder alle Daten in drei Spalten, oder als Liste, dann kann als ID-Code eine ID-Nummer eingegeben werden. »Directory wird ausgedruckt« auch mit Joker, dann werden zum Beispiel alle Daten deren ID-Nummer mit M beginnt nach Eingabe von »M*« ausgedruckt.

```
DRUCK-BEGINN
STAND.KOPF MIT DRUCKEN
PROGRAMM-LISTE DRUCKEN
PREIS-LISTE DRUCKEN
LISTE NACH KRITERIUM
STANDARDKOPF AENDERN

BITTE FUNKTION WAEHLEN
```

Bild 3. Einige Parameter lassen sich bei einem Ausdruck selber bestimmen. So können Sie nicht nur eine Programmliste drucken, sondern auch eine Preisliste. Sie bestimmen, wieviel Zeilen pro Seite und ob der Standardkopf mit ausgedruckt werden soll.

Bild 4. Ein kleines Beispiel einer Programmliste. Hier sind lediglich zwei Disketteninhalte in unsortierter Form zu sehen. Der Kopf kann natürlich weggelassen oder auch geändert werden.▼

F8 gibt die Daten auf den Bildschirm aus.
£ Farbwechsel von Bildschirm und Schrift.
& alle Diskettenoperationen sind möglich.
./. Spezielle Listen können erstellt werden nach dem ersten Zeichen der ID-Nummer (zum Beispiel nach Eingabe von M werden alle Daten deren ID-Nummer mit M beginnt auf ein Datenfile geschrieben.
$ Zeigt die Directory einer Diskette auf dem Bildschirm.
! Erlaubt das Aussortieren fremder Listen wenn die eigene Liste in den Computer geladen wurde. Es werden alle Daten, die in der fremden Liste sind, mit den eigenen nach Kriterium (Anzahl Zeichen von links) durchsucht. Ein in der eigenen Liste nicht vorhandener Datensatz wird dann auf ein Differenzfile abgelegt.

(B. Winkler / gk)

```
100 REM ***********************          <185>
110 REM *                     *          <081>
120 REM * PROGRAMM DISKSORTER *          <202>
130 REM *                     *          <101>
140 REM *  BERNHARD WINKLER   *          <209>
150 REM *                     *          <121>
160 REM *  GIESSENERSTR.42    *          <181>
170 REM *                     *          <141>
180 REM *  5000 KOELN 91      *          <063>
190 REM *                     *          <161>
200 REM ***********************          <029>
210 REM                                  <097>
220 REM                                  <107>
230 PRINT"[CLEAR,DOWN,WHITE]":POKE 53280,0
   :POKE 53281,0                         <203>
240 PRINT"@@[SPACE2]@@@[SPACE]@@@[SPACE]@[SPACE]
   @[SPACE]@@@[SPACE]@@@[SPACE]@@@[SPACE]@@@
   [SPACE]@@@[SPACE]@@@"                 <169>
250 PRINT"@[SPACE]@[SPACE2]@[SPACE2]@[SPACE3]@
   [SPACE]@[SPACE]@[SPACE3]@[SPACE]@[SPACE]@
   [SPACE]@[SPACE2]@[SPACE2]@[SPACE3]@[SPACE]@"
                  <022>
260 PRINT"@[SPACE]@[SPACE2]@[SPACE2]@@@[SPACE]
   @@[SPACE2]@@@[SPACE]@[SPACE]@[SPACE]@@
   [SPACE3]@[SPACE2]@@[SPACE2]@@[SPACE]"  <054>
270 PRINT"@[SPACE]@[SPACE2]@[SPACE4]@[SPACE]@
   [SPACE]@[SPACE3]@[SPACE]@[SPACE]@[SPACE]@
   [SPACE]@[SPACE2]@[SPACE2]@[SPACE3]@[SPACE]@"
                  <043>
```

PRG.NAME	ID	BL.	PRG.NAME	ID	BL.	PRG.NAME	ID	BL
DISKSORTER LADER	BG	3	----ROTH----	BG	1	3D-VIERGEWINNT	00	28
SUPERSORT	BG	5	TASTATURPIEPS	BG	3	AUTO/SIEDEL	00	5
DISKSORTER	BG	57	--FOERTSCH--	BG	1	DEMO KUHN ----	00	1
STANDARDKOPF	BG	1	EIGENES BASIC	BG	44	DEMO1	00	2
DATA ERZEUGER	BG	15	---SCHULZ---	BG	1	DEMO2	00	3
1---------------	BG	1	AUTONUM 64	BG	6	FAST TAPE	00	11
M-P-S	BG	19	SUPER LINE	BG	15	HEX-DATA-AUTOMAT	00	7
ESF EDIT SEQ FIL	BG	21	SUPER LINE DEMO	BG	3	HI--------------	00	1
CHECKSUMMER 64	BG	6	ZEICHEN-EDITOR	BG	18	HIRES-3/1-TEST	00	11
CHECKSUMMER VC20	BG	7	EDITOR	BG	32	HIRES3.B.LADER	00	93
AUTOSTART-64	BG	6	DELETE	BG	6	HIRESMASCH	00	17
AS.ASSEMBLER	BG	10	TRACK 18	BG	41	HIX-------------	00	1
AUTO	BG	6	HILFSROUTINEN	BG	82	LISTSCHUTZ/GAIS	00	7
TOOL-KIT/KUNZ	BG	40	HARDCOPY	BG	6	MERGE/FLEIG	00	8
----HAAS----	BG	1	DISASSEMBLER	BG	16	PET-SIMULATOR	00	4
LADER	BG	1	SINGLE STEP	BG	12	SIMONS-AXO	00	28
DATEI-ORGANISATI	BG	57	LADER VC 20	BG	4	SMON TEIL2.LADER	00	13
DISK-MASTER/M	BG	7	LADER C 64	BG	4	STRINGY	00	34
----FETTE---	BG	1	VERSCHIEBER	BG	14	AHLBORN/SYNTHESI	00	26
BILDSPEICHER	BG	14	MINI GBASIC	BG	34	AHLBORN/SAVEROUT	00	2
OHNE REM	BG	14	!	00	2	TRACE ASSEMBLE	00	36
---RUSSEL---	BG	1	EDITOR	00	20	DTRACE.OBJ	00	10
TASTENDRUCK	BG	12	KREUZWORTRAETSE	00	44	DTRACE.BOBJ	00	10
----DREUW---	BG	1	SUCH.LADER	00	16	MATHE BASIC VC20	00	53
MOUSE 64	BG	10	----------------	00	2			

```
280 PRINT"88[SPACE2]888[SPACE]888[SPACE]8[SPACE]
    8[SPACE]888[SPACE]888[SPACE]8[SPACE]8[SPACE2]
    8[SPACE2]888[SPACE]8[SPACE]8"                <142>
290 PRINT"[DOWN4,RIGHT2]BITTE[SPACE]TASTE[SPACE]
    DRUECKEN!":GOSUB 11000                       <162>
300 REM *** VARIABLEN-DEFINITION ***             <098>
310 REM                                          <198>
320 S$="[DOWN,RIGHT5]":POKE 53280,0:POKE 53281,0
    :PRINT"[GREEN]"                              <228>
340 BF$="BLOCKS[SPACE]FREE":EZ$="[SPACE]ELEMENT"
    :EX$="[SPACE]NICHT"                          <059>
350 V$="[CLEAR,DOWN2,RIGHT2,RVSON,SPACE]"
    :M$="[DOWN2,RIGHT3]":X$="[DOWN2,RIGHT2,RVSON,
    SPACE]":N$="[DOWN,RIGHT,RVSON,SPACE]"
    :ZM$="[RVSON,SPACE]←[SPACE,RVOFF,SPACE,RVSON,
    SPACE2]ZUM[SPACE]MENUE[SPACE]"               <108>
360 FI$="STANDARDKOPF"                           <147>
370 REM                                          <002>
380 REM *** STANDARDKOF EINLESEN ***             <140>
390 REM                                          <022>
400 OPEN 7,8,7,FI$+",S,R"                        <248>
410 INPUT#7,BN$,BA$,BW$,BT$:CLOSE 7
    :PRINT X$;"DATUM[SPACE]?";:GOSUB 18000:DA$=K$
                       <112>
420 DIM B$(850),FF$(200):OPEN 3,8,15             <064>
430 REM                                          <062>
440 REM *** MENUE ***                            <190>
450 N$="[DOWN,RIGHT,RVSON,SPACE]"                <107>
460 ZK$="[SPACE26]"                              <140>
470 PRINT N$;"[CLEAR,RIGHT,SPACE](C)1983[SPACE,
    RVOFF,SPACE]BY[SPACE]B.WINKLER[SPACE]
    TEL(0221)881373"                             <108>
480 PRINT N$;"F1[SPACE,RVOFF,SPACE]DATEN[SPACE]
    PER[SPACE]DISK[SPACE]EINLESEN"               <069>
490 PRINT N$;"F3[SPACE,RVOFF,SPACE]DATEN[SPACE]
    ERWEITERN[SPACE]/[SPACE]LOESCHEN[SPACE]"
                       <033>
500 PRINT N$;"F5[SPACE,RVOFF,SPACE]LISTE[SPACE]
    EDITIEREN"                                   <166>

510 PRINT N$;"F7[SPACE,RVOFF,SPACE]LISTE[SPACE]
    SORTIEREN"                                   <212>
520 PRINT N$;"F2[SPACE,RVOFF,SPACE]DATEN[SPACE]
    ABSPEICHERN"                                 <046>
530 PRINT N$;"F4[SPACE,RVOFF,SPACE]DATEN[SPACE]
    EINLESEN[SPACE4,RVSON,SPACE]CLR[SPACE,RVOFF,
    SPACE]QUIT"                                  <049>
540 PRINT N$;"F6[SPACE,RVOFF,SPACE]LISTE[SPACE]
    DRUCKEN[SPACE5,RVSON,SPACE]%[SPACE,RVOFF,
    SPACE]SPEZ.LISTE"                            <253>
550 PRINT N$;"F8[SPACE,RVOFF,SPACE]LISTE[SPACE]
    AUF[SPACE]SCREEN[SPACE2,RVSON,SPACE]&[SPACE,
    RVOFF,SPACE]DISK[SPACE]OPER."                <056>
560 PRINT N$;"*[SPACE2,RVOFF,SPACE]";BF$;"
    [SPACE7,RVSON,SPACE]$[SPACE,RVOFF,SPACE]
    DIRECTORY[SPACE]""                           <125>
570 PRINT N$;"£[SPACE2,RVOFF,SPACE]ANDERE[SPACE]
    BILDSCHIRM[SPACE]FARBE"                      <164>
580 PRINT N$;"![SPACE2,RVOFF,SPACE]AUSSORTIEREN
    [SPACE]FREMDER[SPACE]LISTE[SPACE]"           <173>
590 GOSUB 11000                                  <207>
600 IF A$=CHR$(133)THEN I=0:BB=B:B=0:Q=0
    :GOSUB 1000:IF A$="←"THEN B=BB              <032>
610 IF A$=CHR$(138)THEN GOSUB 6000               <180>
620 IF A$=CHR$(37)THEN GOSUB 8200                <144>
630 IF A$=CHR$(135)THEN OPEN 4,4:GOSUB 7000
    :CLOSE 4                                     <065>
640 IF A$=CHR$(38)THEN GOSUB 6300                <164>
650 IF A$=CHR$(33)THEN GOSUB 14000               <213>
660 IF A$=CHR$(36)THEN B=B+2:I=B:H=1:GT=1:G=0
    :GOSUB 1050:H=0:B=B-2:GT=0                   <094>
670 IF A$=CHR$(136)THEN GOSUB 4000               <236>
680 IF A$=CHR$(147)THEN SYS 64738                <081>
690 IF A$=CHR$(137)THEN GOSUB 5000               <002>
700 IF A$=CHR$(134)THEN GOSUB 9000               <013>
710 IF A$=CHR$(139)THEN OPEN 4,4:GOSUB 3000
    :CLOSE 4                                     <145>
720 IF A$=CHR$(140)THEN GOSUB 8000               <029>
730 IF A$=CHR$(92)THEN GOSUB 2000                <247>
740 IF A$=CHR$(42)THEN BB=B:GOSUB 12000:B=BB
                       <145>
750 GOTO 450                                     <018>
760 REM                                          <137>
1000 REM                                         <122>
1010 REM *** DISKLESER ***                       <038>
1020 REM                                         <142>
1030 G=0:PRINT V$;"DISK[SPACE]EINLEGEN[SPACE]
    :[SPACE]DATEN[SPACE]KOMPLETT[SPACE]"
    :PRINT X$;"ODER[SPACE]EINZELN[SPACE]";       <207>
1035 PRINT"EINLESEN[SPACE]?[SPACE](K/E)"
    :PRINT X$;ZM$:GOSUB 10000:IF A$="E"THEN G=1
                       <209>
1040 IF A$="←"THEN RETURN                        <142>
1050 PRINT V$;"ICH[SPACE]LESE[SPACE]DISK[SPACE]"
    :PRINT X$;                                   <098>
1060 OPEN 1,8,0,"$0" :GB=1                       <034>
1070 GOSUB 16000:PRINT"[SPACE,RVSON]"K$"[DOWN2]"
    :IF T<>0 THEN GOSUB 10000:CLOSE 1:RETURN
                       <089>
1080 GET#1,A$,F$                                 <088>
1090 GET#1,A$,F$:B$(I+1)=""                      <109>
1100 GET#1,A$,F$                                 <108>
1110 C=0:L$="[SPACE3]"                           <031>
1120 IF A$<>""THEN C=ASC(A$)                     <020>
1130 IF F$<>""THEN C=C+ASC(F$)*256               <094>
1140 Q$=MID$(STR$(C),2):L$=LEFT$(Q$+L$,3)        <211>
1150 GET#1,F$:IF F$=""THEN PRINT"[DOWN,RIGHT2,
    RVSON]";L$;"[SPACE]";BF$:GOTO 1295           <184>
1160 IF F$<>CHR$(34)THEN 1150                    <210>
1170 GET#1,F$:IF F$<>CHR$(34)THEN B$(I+Q)=B$(I+
    Q)+F$:GOTO 1170                              <034>

1180 IF Q=1 THEN B$(I+1)=""                      <215>
1190 GET#1,F$:IF F$=CHR$(32)THEN 1190            <005>
1200 C$=""                                       <017>
1210 C$=C$+F$:GET#1,F$:IF F$<>""THEN 1210        <089>
1220 IF GB=1 THEN GB$=C$:GB=0                    <209>
1230 B$(I+Q)=LEFT$(B$(I+Q)+ZK$,17)+LEFT$(GB$,
    3)+RIGHT$(ZK$+Q$,3)                          <116>
1240 PRINT"[RIGHT2]";B$(I+Q);"[RIGHT3]";"[RVSON,
    SPACE]";LEFT$(C$,5);:IF G=0 AND GT=0 THEN PR
    INT                                          <065>
1250 IF GT=1 THEN GT=0:B$(I)="":PRINT:GOTO 1090
                       <163>
1260 IF Q=1 THEN Q=0:B$(I+1)=""                  <148>
1270 IF G=1 AND GT=0 AND I>0 THEN PRINT"[RVOFF,
    SPACE]J/N[LEFT3]";:GOSUB 11000
    :IF A$="N"THEN PRINT"NEIN";:GT=1:GOTO 1250
                       <246>
1280 I=I+1:IF G=1 THEN PRINT"[RVOFF,SPACE]JA"
                       <180>
1285 IF I+Q+1=850 THEN PRINT X$;"SPEICHER[SPACE]
    IST[SPACE]VOLL":PRINT X$;ZM$:CLOSE 1
    :GOSUB 11000:RETURN                          <159>
1290 GOTO 1090                                   <098>
1295 CLOSE 1:IF H=1 THEN GOSUB 10000:RETURN
                       <194>
1300 PRINT X$;"NOCH[SPACE]EINE[SPACE]DISK[SPACE]
    (J/N)"                                       <057>
1310 QW$=STR$(I-1)                               <219>
1320 PRINT X$;"BISHER[SPACE]WURDEN[SPACE]";QW$;
    "[SPACE]DATEN[SPACE]GELESEN[SPACE,DOWN4]"
                       <140>
1330 GOSUB 11000                                 <182>
1340 IF A$="J"THEN GT=1:PRINT V$;:GOTO 1060
                       <168>
1350 IF A$<>"N"THEN 1330                         <159>
1360 B=I-1:CLOSE 1                               <121>
1370 A$="":T=0                                   <042>
1380 RETURN                                      <247>
2000 REM                                         <102>
2010 REM *** FARBWECHSEL ***                     <146>
2020 REM                                         <122>
2030 I=0:F=0:J=0                                 <232>
2040 PRINT V$;"F1[RVOFF,SPACE,RVSON]FARBWECHSEL
    [SPACE]BILDSCHIRM"                           <179>
2050 PRINT X$;"F3[RVOFF,SPACE,RVSON]FARBWECHSEL
    [SPACE]HINTERGRUND"                          <043>
2060 PRINT X$;"F5[RVOFF,SPACE,RVSON]FARBWECHSEL
    [SPACE]SCHRIFT"                              <000>
2070 PRINT X$;ZM$                                <057>
2080 GOSUB 11000                                 <135>
2090 IF A$=CHR$(133)THEN I=I+1:POKE 53280,I
                       <145>
2100 IF A$=CHR$(134)THEN J=J+1:POKE 53281,J
                       <160>
2110 IF A$=CHR$(135)THEN F=F+1:POKE 646,F        <060>
2120 IF A$="←"THEN RETURN                        <202>
2130 GOTO 2040                                   <169>
3000 REM                                         <082>
3010 REM *** DRUCKER AUSGABE ***                 <096>
3020 ZK$=LEFT$(ZK$+ZK$,33)                       <057>
3030 J$="[RVSON,SPACE]JA[SPACE]"
    :N$="[RVSON]NEIN":F1$=N$:F3$=J$:F5$=J$:F7$=N$
```

```
   :F2$=N$:Z$="[RIGHT5]"                             <225>
3040 Y$="[DOWN]"+Z$:F4$=N$:DR=1                      <070>
3050 REM                                             <132>
3060 T$="[RIGHT,RVSON,SPACE]"                        <156>
3070 ST$="":AN$="BITTE[SPACE]FUNKTION[SPACE]
   WAEHLEN"                                          <151>
3075 PRINT"[CLEAR,RIGHT5,RVSON,SPACE]DRUCKER
   [SPACE]AUSGABE[SPACE,DOWN]"                       <059>
3080 PRINT F1$;T$;"F1[SPACE,RVOFF,SPACE]
   DRUCK-BEGINN[SPACE,DOWN]"                         <237>
3090 PRINT F3$;T$;"F3[SPACE,RVOFF,SPACE]
   STAND.KOPF[SPACE]MIT[SPACE]DRUCKEN[SPACE,
   DOWN]"                                            <112>

3100 PRINT F5$;T$;"F5[SPACE,RVOFF,SPACE]
   PROGRAMM-LISTE[SPACE]DRUCKEN[SPACE,DOWN]"
                       <207>
3110 PRINT F7$;T$;"F7[SPACE,RVOFF,SPACE]
   PREIS-LISTE[SPACE]DRUCKEN[SPACE,DOWN]"   <251>
3120 PRINT F2$;T$;"F2[SPACE,RVOFF,SPACE]LISTE
   [SPACE]NACH[SPACE]KRITERIUM[SPACE,DOWN]"
                       <021>
3130 PRINT F4$;T$;"F4[SPACE,RVOFF,SPACE]
   STANDARDKOPF[SPACE]AENDERN[SPACE]"                <057>
3140 PRINT X$;"[LEFT2]";ZM$                          <036>
3150 PRINT"[DOWN2]"                                  <089>
3160 PRINT"[SPACE4]*******************************
   *******"                                          <061>
3170 PRINT"[SPACE4]*[SPACE33]*"                      <173>
3180 PRINT"[SPACE4]*******************************
   *******"                                          <081>
3190 PRINT"[SPACE4]*[SPACE33]*"                      <193>
3200 PRINT"[SPACE4]*******************************
   *******"                                          <113>
3210 PRINT"[HOME]";:FOR J=1 TO 20:PRINT"[DOWN]";
   :NEXT:PRINT Z$;;ST$                               <096>
3220 PRINT Y$;AN$                                    <164>
3230 IF C=1 THEN C=0:GOSUB 3450:F4$=N$:GOTO 3070
                       <067>
3240 IF F1$=J$THEN F1$=N$:GOSUB 3380:GOTO 3070
                       <130>
3250 GOSUB 11000                                     <061>
3260 IF A$=CHR$(133)AND F1$=N$THEN F1$=J$
   :GOTO 3070                                        <118>
3270 IF A$=CHR$(134)AND F3$=J$THEN F3$=N$
   :GOTO 3070                                        <133>
3280 IF A$=CHR$(134)AND F3$=N$THEN F3$=J$
   :GOTO 3070                                        <143>
3290 IF A$=CHR$(135)AND F5$=N$THEN F5$=J$:F2$=N$
   :F7$=N$:GOTO 3070                                 <151>
3300 IF A$=CHR$(135)AND F5$=J$THEN F5$=N$
   :GOTO 3070                                        <168>
3310 IF A$=CHR$(136)AND F7$=J$THEN F7$=N$
   :GOTO 3070                                        <183>
3320 IF A$=CHR$(136)AND F7$=N$THEN F7$=J$:F2$=N$
   :F5$=N$:GOTO 3070                                 <184>
3330 IF A$=CHR$(137)AND F2$=N$THEN F2$=J$:F7$=N$
   :F5$=N$:GOTO 3070                                 <191>
3340 IF A$=CHR$(137)AND F2$=J$THEN F2$=N$
   :GOTO 3070                                        <205>
3350 IF A$=CHR$(95)THEN DR=0:RETURN                  <050>
3360 IF A$=CHR$(138)THEN F4$=J$:C=1:GOTO 3070
                       <181>
3370 IF A$<CHR$(95)OR A$>CHR$(137)THEN 3250
                       <074>
3380 PRINT"[HOME]":FOR J=1 TO 19:PRINT"[DOWN]";
   :NEXT:PRINT Z$;ZK$:PRINT"[UP]";Z$;"OK[SPACE]
   ?[SPACE](J/N)"                                    <108>
3385 PRINT Y$;ZK$:GOSUB 11000
   :IF A$<>"J"THEN RETURN                            <106>
3390 PRINT"[HOME]":FOR J=1 TO 19:PRINT"[DOWN]";
   :NEXT:PRINT Z$;"WIEVIEL[SPACE]ZEILEN[SPACE]
   PRO[SPACE]SEITE[SPACE]?>=15"                      <135>
3400 PRINT Y$;ZK$:PRINT"[UP2]";Y$;:GOSUB 18000
   :DZ=VAL(K$):IF DZ<15 OR DZ>300 THEN 3390
                       <238>
3405 IF F2$=N$AND F5$=N$AND F7$=N$THEN RETURN
                       <192>
3410 IF F3$=J$THEN 3530                              <025>
3420 IF F3$=N$THEN 3590                              <045>
3430 IF A$="←"THEN RETURN                            <237>
3440 RETURN                                          <011>
3450 ST$="DATUM[SPACE]?":GOSUB 3510:DA$=K$ <150>
3460 ST$="NAME[SPACE]?":GOSUB 3510:BN$=K$  <081>
3470 ST$="STRASSE,NR.[SPACE]?":GOSUB 3510:BA$=K$
                       <076>
3480 ST$="WOHNORT[SPACE]?":GOSUB 3510:BW$=K$
                       <126>
3490 ST$="TELEFON[SPACE]?":GOSUB 3510:BT$=K$
                       <097>
3500 RETURN                                          <071>
3510 PRINT"[HOME]":FOR J=1 TO 19:PRINT"[DOWN]";
   :NEXT:PRINT Z$;ST$                                <031>
3515 PRINT Y$;ZK$:PRINT"[UP2]";Y$;:GOSUB 18000
                       <206>
3520 PRINT"[HOME]":FOR J=1 TO 19:PRINT"[DOWN]";
   :NEXT:PRINT Z$;ZK$:PRINT Y$;ZK$:RETURN  <067>
3530 ZD=0:IF F3$=J$THEN ZD=15                        <054>
3540 PRINT#4,CHR$(14)"[SPACE2]";BN$
   :PRINT#4,"[SPACE2]";BA$                           <225>
3550 PRINT#4,"[SPACE2]";BW$:PRINT#4,"[SPACE2]";
   BT$:PRINT#4                                       <144>
3560 PRINT#4,"[SPACE2]DISKETTENINHALT[SPACE]VOM
   [SPACE]";DA$                                      <114>
3570 PRINT#4:PRINT#4,"[SPACE2]PROGRAMME
   :";B;CHR$(15)                                     <091>
3580 PRINT#4                                         <213>
3590 IF F2$=J$THEN 3760                              <210>
3595 IF F7$=N$THEN 3610                              <218>
3600 QS$="PRG.NAME[SPACE8]KB.[SPACE]PREIS[SPACE]
   ":GOTO 3620                                       <224>
3610 QS$="PRG.NAME[SPACE9]ID[SPACE]BL.[SPACE3]"
                       <103>
3620 ZD$="[SPACE2]":GOTO 3640                        <122>
3640 DD=INT(B/3)+1                                   <131>
3650 S=1                                             <134>
3660 B$(B+1)="":B$(B+2)="":B$(B+3)=""                <047>
3670 PRINT#4,LEFT$("[SPACE4]"+QS$+QS$+QS$,78)
                       <170>
3680 PRINT#4,"[SPACE4]";:FOR T=1 TO 74
   :PRINT#4,CHR$(101);:NEXT:PRINT#4                  <191>
3690 FOR T=1 TO DD                                   <092>
3700 PRINT#4,"[SPACE4]";B$(S);"[SPACE3]";
   B$(S+DD);"[SPACE3]";B$(S+2*DD)                    <205>
3710 ZD=ZD+1                                         <085>
3720 S=S+1:IF ZD=DZ THEN ZD=0:GOSUB 3740             <056>
3730 NEXT:RETURN                                     <234>
3740 PRINT"[HOME]":FOR J=1 TO 19:PRINT"[DOWN]";
   :NEXT:PRINT Z$;                                   <059>
3750 PRINT"DRUCKERPAPIER[SPACE]JUSTIEREN[SPACE]
   ?[SPACE]":PRINT Y$;"TASTE[SPACE]DRUECKEN
   [SPACE]";:GOSUB 11000:RETURN                      <183>
3760 F=2:PRINT"[HOME]":FOR I=1 TO 19
   :PRINT"[DOWN]";:NEXT:PRINT Z$;ZK$
   :PRINT Y$;ZK$:PRINT"[UP4]"                        <042>
3765 PRINT Z$;"ID.CODE[SPACE]EINGEBEN[SPACE]!"
   :PRINT Y$;:GOSUB 18000                            <216>
3770 IF RIGHT$(K$,1)="*"THEN F=1                     <041>
3780 K$=LEFT$(K$+ZK$,F)                              <096>
3790 FOR I=1 TO B:IF ZD=DZ THEN ZD=0:GOSUB 3740
                       <222>
3800 IF MID$(B$(I),18,F)=K$THEN PRINT#4,"
   [SPACE4]";B$(I):ZD=ZD+1                           <213>
3810 NEXT:PRINT#4                                    <120>
3820 RETURN                                          <136>
4000 REM                                             <062>
4010 REM *** SORTIERROUTINE ***                      <146>
4020 REM                                             <082>
4030 PRINT V$;"SORTIER-ROUTINE[SPACE]LAEUFT!"
   :B$(B+1)="ZZZZZZZZ":F=0                           <242>
4035 PRINT X$;"ALPHABETISCH[SPACE]ODER[SPACE]
   NACH[SPACE]ID.NR.[SPACE]?"                        <218>
4040 PRINT X$;"(A/I)":PRINT X$;ZM$:GOSUB 11000
   :IF A$="←"THEN RETURN                             <055>
4055 IF A$="I"THEN F=1                               <229>
4060 FOR C=1 TO B                                    <120>
4070 J=C+1:PRINT X$;"ELEMENT[SPACE7,LEFT4]";J-1;
   "[UP3]"                                           <238>
4080 FOR D=J TO B                                    <166>
4090 IF F=1 AND RIGHT$(B$(C),6)>RIGHT$(B$(D),
   6)THEN 4120                                       <120>
4100 IF B$(C)>B$(D)THEN 4120                         <179>
4110 GOTO 4130                                       <111>
4120 C$(0)=B$(C):B$(C)=B$(D):B$(D)=C$(0)             <108>
4130 NEXT                                            <180>
4140 NEXT                                            <190>
4150 PRINT V$;"DOPPELTE[SPACE]DATEN[SPACE]
   AUSSORTIEREN[SPACE](J/N)[SPACE]"                  <093>
4160 GOSUB 11000                                     <207>
4170 IF A$="J"THEN 4190                              <255>
4180 RETURN                                          <242>
4190 PRINT V$;"ES[SPACE]WIRD[SPACE]AUSSORTIERT
   [SPACE]":J=1                                      <154>
4200 PRINT X$;"WIEVIELE[SPACE]ZEICHEN[SPACE]
```

```
    BERUECKSICHTIGEN[SPACE]?":PRINT X$;
    :GOSUB 18000                              <148>
4210 SS=VAL(K$):IF SS=0 OR SS>25 THEN SS=25
                        <128>
4220 IF LEFT$(B$(J),SS)=LEFT$(B$(J+1),
    SS)THEN 4270                              <240>
4230 J=J+1                                    <183>
4240 IF J<=B-1 THEN 4220                      <103>
4250 FOR I=B+1 TO B+10:B$(I)="":NEXT          <032>
4260 RETURN                                   <066>
4270 FOR I=J TO B                             <106>
4280 B$(I)=B$(I+1)                            <085>
4290 NEXT                                     <084>
4300 J=J-1:B=B-1:GOTO 4230                    <214>
4310 RETURN                                   <116>
5000 REM                                      <042>
5010 REM *** DATEN FILE SPEICHERN ***         <093>
5020 REM                                      <062>
5030 PRINT V$;"DATEN[SPACE]WERDEN[SPACE]
    GESPEICHERT[SPACE]"                       <175>
5040 GOSUB 17000:I=0:PRINT X$;"SICHER[SPACE]?
    [SPACE](J/N)":GOSUB 11000
    :IF A$<>"J"THEN RETURN                    <154>
5050 OPEN 2,8,2,FI$+",S,W"                    <047>
5060 GOSUB 16000:PRINT X$;K$
    :IF T<>0 THEN CLOSE 2:GOSUB 10000:RETURN
                        <250>
5070 PRINT#2,B                                <025>
5080 IF G=1 THEN FOR I=1 TO Z:PRINT#2,FF$(I)
    :GOTO 5100                                <245>
5090 FOR I=1 TO B:PRINT#2,B$(I)               <184>
5100 NEXT                                     <129>
5110 CLOSE 2                                  <219>
5120 RETURN                                   <162>
6000 REM                                      <022>
6010 REM *** DATEN-FILE EINLESEN ***          <040>
6020 REM                                      <042>
6030 PRINT V$;"DATEN[SPACE]WERDEN[SPACE]GELESEN
    [SPACE]":PRINT X$;ZM$                     <192>
6040 GOSUB 17000:IF K$="←"THEN RETURN         <242>
6050 PRINT X$;"DATEN[SPACE]FILE[SPACE]";FI$"
    [SPACE]EINLESEN[SPACE](J/N)[SPACE]"       <118>
6060 GOSUB 11000                              <066>
6070 IF A$<>"J"THEN RETURN                    <228>
6080 OPEN 2,8,2,FI$+",S,R":PRINT"[DOWN2,RIGHT3]
    ";                                        <255>
6090 GOSUB 16000:PRINT"[SPACE,RVSON]"K$"[DOWN2]"
    :IF T<>0 THEN GOSUB 10000:CLOSE 2:RETURN
                        <009>
6100 INPUT#2,B                                <015>
6110 FOR I=FJ+1 TO B+FJ                       <252>
6120 INPUT#2,B$(I):NEXT:CLOSE 2:B=B+FJ:RETURN
                        <027>
6300 PRINT V$;"DISK[SPACE]OPERATION[SPACE]MODE"
                        <071>
6310 PRINT X$;"DISKETTENBEFEHLE[SPACE]S,[SPACE]
    V,[SPACE]N,[SPACE]R,[SPACE]I[SPACE]"      <042>
6320 PRINT X$;"BEFEHL[SPACE]?";:GOSUB 18000
    :FI$=K$:PRINT X$;"AUSFUEHREN[SPACE](J/N)
    [SPACE]?"                                 <101>
6325 GOSUB 11000                              <076>
6330 IF A$="←"THEN RETURN                     <076>
6340 IF A$="J"THEN OPEN 15,8,15:PRINT#15,FI$
    :CLOSE 15:GOSUB 16000:IF T<>0 THEN PRINT X$;
    K$:GOSUB 10000                            <142>
6345 IF T=0 THEN RETURN                       <215>
6350 PRINT"[HOME]";X$:GOTO 6310               <026>
7000 REM                                      <002>
7010 REM *** EDITOR ***                       <207>
7020 REM                                      <022>
7030 POKE 650,128                             <138>
7040 PRINT S$;"[CLEAR,RIGHT5,RVSON,SPACE]@
    [SPACE,RVOFF,SPACE2]:[SPACE2]ELEMENT[SPACE]
    LOESCHEN"                                 <199>
7050 PRINT S$;"[RVSON,SPACE]£[SPACE,RVOFF,
    SPACE2]:[SPACE2]ELEMENT[SPACE]SUCHEN"     <062>
7060 PRINT S$;"[RVSON,SPACE]*[SPACE,RVOFF,
    SPACE2]:[SPACE2]ELEMENT[SPACE]VERBESSERN"
                        <079>
7070 PRINT S$;"[RVSON,SPACE]←[SPACE,RVOFF,
    SPACE2]:[SPACE2]";RIGHT$(ZM$,13)          <053>
7080 PRINT S$;"[RVSON,SPACE]P[SPACE,RVOFF,
    SPACE2]:[SPACE2]ELEMENT[SPACE]DRUCKEN[SPACE]"
                        <150>
7090 PRINT S$;"[RVSON,SPACE3,RVOFF,SPACE2]
    :[SPACE2]EIN[SPACE]ELEMENT[SPACE]VOR"     <023>
7100 PRINT S$;"[RVSON,SPACE]F1[RVOFF,SPACE2]
    :[SPACE2]20[SPACE]ELEMENTE[SPACE]VOR[SPACE]"
                        <099>
7110 PRINT S$;"[RVSON,SPACE]F3[RVOFF,SPACE2]
    :[SPACE2]20[SPACE]ELEMENTE[SPACE]ZURUECK"
                        <161>
7120 PRINT S$;"[RVSON,SPACE]↑[SPACE,RVOFF,
    SPACE2]:[SPACE2]EIN[SPACE]ELEMENT[SPACE]
    ZURUECK"                                  <197>
7130 PRINT S$;"[RVSON,SPACE]$[SPACE,RVOFF,
    SPACE2]:[SPACE2]DM-LISTE[SPACE]ERSTELLEN"
                        <115>
7135 PRINT S$;"[RVSON,SPACE]+[SPACE,RVOFF,
    SPACE2]:[SPACE2]ELEMENT[SPACE]EINFUEGEN
    [DOWN2,SPACE]":PRINT                      <039>
7140 FOR I=1 TO B                             <146>
7150 PRINT"[UP,RIGHT3]";ZK$;"[DOWN]"
    :PRINT"[RIGHT3,SPACE2]";B$(I)             <011>
7160 GOSUB 11000                              <146>
7170 IF B$(1)=""AND B>1 THEN A$="@":I=1       <002>
7180 IF A$="@"AND B>1 THEN B$(I)=B$(B):B=B-1
    :A$=""                                    <084>
7190 IF A$=CHR$(133)AND I+20<B THEN I=I+20    <115>
7200 IF A$=CHR$(134)AND I-20>0 THEN I=I-20    <076>
7210 IF A$="*"THEN PRINT"[UP2,RIGHT5]";"[RVSON]
    KORREKTURELEMENT":INPUT"[RIGHT3]";B$(I)   <134>
7220 IF A$="*"THEN PRINT"[UP2]";ZK$:PRINT
    :IF B$(I)=""THEN B$(I)="ZZZZ"             <157>
7225 IF A$="$"THEN GOSUB 7500:GOTO 7000       <233>
7230 IF A$="←"THEN A$="":POKE 650,0:RETURN    <049>
7240 IF A$="↑"THEN I=I-2:IF I<1 THEN I=B-1    <034>
7250 IF A$="+"THEN B=B+1:PRINT"[RVSON,UP2,
    RIGHT5]NEUES[SPACE]ELEMENT"
    :INPUT"[RIGHT3]";B$(B):PRINT"[UP2]";ZK$:PRINT
                        <059>
7260 IF A$="P"THEN PRINT#4,"[SPACE4]";B$(I)
                        <204>
7270 IF A$="£"THEN PRINT V$;"NACH[SPACE]WELCHEM
    [SPACE]ELEMENT[SPACE]SUCHEN[SPACE]?"
    :PRINT X$;:GOSUB 18000:GOSUB 7310:GOTO 7000
                        <145>
7280 IF LEFT$(B$(B),4)="ZZZZ"AND I>1 THEN B=B-1
                        <122>
7290 PRINT"[UP3,RIGHT3]";ZK$:NEXT:GOTO 7140
                        <204>

7300 RETURN                                   <046>
7310 IF K$=""THEN RETURN                      <207>
7320 F=0:IF RIGHT$(K$,1)="*"THEN F=1          <119>
7330 Q=LEN(K$)-F:Q$=LEFT$(K$,Q):EZ=0          <086>
7340 EE$="[SPACE]KEIN[SPACE]WEITERES"
    :EY$="[SPACE]VORHANDEN[SPACE]":PRINT      <101>
7350 FOR K=1 TO B                             <103>
7360 IF LEFT$(B$(K),Q)=Q$THEN EZ=1:SE=1:R=1
                        <115>
7370 IF SE=0 THEN NEXT                        <020>
7380 IF SE=1 THEN PRINT"[RIGHT8]";B$(K):SE=0
    :NEXT                                     <015>
7390 IF EZ=0 THEN PRINT"[RIGHT2,DOWN,RVSON]";
    EZ$+EX$+EY$                               <188>
7400 IF EZ=1 THEN PRINT"[DOWN2]"
    :PRINT"[RIGHT2,RVSON]";EE$+EZ$+EY$
    :PRINT"[RIGHT2]";WS$                      <054>
7410 PRINT X$;ZM$:GOSUB 11000                 <226>
7420 RETURN                                   <166>
7500 PRINT"[CLEAR,DOWN2,RIGHT2,RVSON,SPACE]
    DM-LISTENERSTELLUNG[SPACE,RVOFF,DOWN2]"
    :PRINT"[DOWN2]";X$;ZM$:PRINT"[HOME,DOWN5]"
                        <002>
7510 FOR J=I TO B:PRINT"[SPACE2]";
    :C$=LEFT$(B$(J),17):PRINT C$;             <231>
7520 F$=RIGHT$(ZK$+STR$(INT(VAL(RIGHT$(B$(J),
    3))/4+.5)),2):PRINT F$;                   <149>
7530 PRINT"[SPACE]000[LEFT3]";:GOSUB 18000
    :IF K$="←"THEN RETURN                     <132>
7540 B$(J)=C$+F$+"[SPACE]"+LEFT$(K$+ZK$,3)    <177>
7550 PRINT"[UP]";:NEXT:RETURN                 <200>
8000 REM                                      <238>
8010 REM *** BILDSCHIRM-AUSGABE ***           <250>
8020 REM                                      <002>
8030 PRINT V$;"INHALT[SPACE]EINER[SPACE]DISK
    [SPACE]LISTEN[SPACE]":PRINT X$;"ODER[SPACE]
    ALLE[SPACE]DISK[SPACE]DATEN[SPACE]LISTEN
    [SPACE](E/A)[SPACE]"                      <190>
8040 GOSUB 11000                              <006>
8050 :IF A$="E"THEN 8120                      <104>
8060 PRINT V$;ZM$:PRINT:Z=1                   <254>
8070 FOR I=1 TO B:Z=Z+1                       <179>
```

```
8080 PRINT"[RIGHT8]";B$(I):IF PEEK(631)=95 THEN
     POKE 198,0:RETURN                          <069>
8090 IF Z=23 THEN GOSUB 11000:Z=1
     :IF H=1 THEN RETURN                        <119>
8100 NEXT:GOSUB 10000                           <253>
8110 RETURN                                     <091>
8120 PRINT V$;"WELCHE[SPACE]DISK[SPACE]LISTEN
    [SPACE]":PRINT X$;"ID.NR[SPACE]EINGEBEN
    [SPACE]";:GOSUB 18000                       <124>
8130 FOR I=1 TO B:IF K$=MID$(B$(I),18,
     2)THEN PRINT"[RIGHT8]";B$(I)               <015>
8140 NEXT                                       <109>
8150 GOSUB 10000                                <115>
8160 PRINT X$;"NOCH[SPACE]EINE[SPACE]DISK[SPACE]
    LISTEN[SPACE](J/N)[SPACE]":GOSUB 11000
    :IF A$="J"THEN 8120                         <131>
8170 RETURN                                     <151>
8200 REM                                        <151>
8210 REM *** LISTEN NACH ID KRITERIUM ERSTELLEN
     ***                                        <125>
8220 REM                                        <171>
8230 PRINT V$;"LISTEN[SPACE]NACH[SPACE]ID[SPACE]
    KRITERIUM[SPACE]ERSTELLEN[SPACE]"
    :PRINT X$;ZM$                               <237>
8240 PRINT X$;"NACH[SPACE]WELCHEM[SPACE]
    KRITERIUM[SPACE]SUCHEN[SPACE]?";:GOSUB 18000
    :F$=LEFT$(K$,1)                             <218>

8250 IF F$="←"THEN RETURN                       <185>
8260 PRINT X$;"DATENDISKETTE[SPACE]EINLEGEN
    [SPACE]":GOSUB 17000:FI$="@:"+FI$           <097>
8270 X=0:FOR I=1 TO B:IF MID$(B$(I),18,
     1)=F$THEN X=X+1                            <039>
8280 NEXT                                       <218>
8290 IF X=0 THEN PRINT X$;"KEIN[SPACE]ELEMENT
    [SPACE]VORH.":GOSUB 10000:RETURN            <128>
8300 OPEN 1,8,1,FI$+",S,W":PRINT#1,X            <083>
8310 GOSUB 16000:IF T<>0 THEN GOSUB 10000
     :CLOSE 1:RETURN                            <159>
8320 FOR I=1 TO B:IF MID$(B$(I),18,
     1)=F$THEN PRINT"[SPACE8]";B$(I):PRINT#1,B$(I)
                  <239>
8330 NEXT:CLOSE 1:RETURN                        <223>
9000 REM                                        <218>
9010 REM *** DATEN ERWEITER/LOESCHEN ***        <051>
9020 REM                                        <238>
9030 PRINT V$;"F1[SPACE,RVOFF,SPACE]DATEN[SPACE]
    ERWEITERN[SPACE]"                           <037>
9040 PRINT X$;"F3[SPACE,RVOFF,SPACE]DATEN[SPACE]
    FILE[SPACE]ZULADEN                          <143>
9050 PRINT X$;"F5[SPACE,RVOFF,SPACE]DATEN[SPACE]
    LOESCHEN[SPACE](ID.NR.)[SPACE]"             <181>
9060 PRINT X$;"F7[SPACE,RVOFF,SPACE]
    STANDARDKOPF[SPACE]SPEICHERN[SPACE]"        <076>
9070 PRINT X$;ZM$                               <172>
9080 GOSUB 11000                                <026>
9090 IF A$=CHR$(95)THEN RETURN                  <002>
9100 IF A$=CHR$(133)THEN GT=1:I=B+1:GOTO 1000
                  <251>
9110 IF A$=CHR$(134)THEN FJ=B:GOSUB 6000:FJ=0
     :RETURN                                    <055>
9120 IF A$=CHR$(135)THEN 9160                   <141>
9130 IF A$=CHR$(136)THEN OPEN 7,8,7,"@
     :STANDARDKOPF,S,W":PRINT#7,BN$:PRINT#7,BA$
                  <151>
9140 IF A$=CHR$(136)THEN PRINT#7,BW$:PRINT#7,BT$
     :CLOSE 7                                   <138>
9150 GOTO 9030                                  <054>
9160 PRINT V$;"EINGABE[SPACE]DER[SPACE]ZU[SPACE]
    LOESCHENDEN[SPACE]ID[SPACE]NR.[SPACE]?[SPACE]
    ":PRINT X$;                                 <062>
9170 GOSUB 18000                                <123>
9180 PRINT X$;"SOLL[SPACE]";K$;"[SPACE]
    GELOESCHT[SPACE]WERDEN[SPACE](J/N)[SPACE]?
    [SPACE]"                                    <176>
9190 GOSUB 11000:IF A$<>"J"THEN RETURN          <217>
9200 FOR I=1 TO B                               <166>
9210 IF K$=MID$(B$(I),18,2)THEN GOSUB 9240      <218>
9220 NEXT                                       <170>
9230 RETURN                                     <192>
9240 FOR J=I TO B:B$(J)=B$(J+1):NEXT            <109>
9250 I=I-1:B=B-1:RETURN                         <122>
10000 PRINT X$;"TASTE[SPACE]DRUECKEN[SPACE]":T=0
      :GOSUB 11000:RETURN                       <142>
11000 GET A$:IF A$=""THEN 11000                 <225>
11010 RETURN                                    <187>
12000 REM                                       <157>
12010 REM *** BLOCKS FREE ROUTINE ***           <169>
12020 REM                                       <177>
12030 PRINT V$;BF$;"[SPACE]FUNKTION"            <019>
12040 FI$="@:"+BF$:FOR I=1 TO 200:FF$(I)="":NEXT
                  <003>
12050 FF=1:K=2:I=2:BR$="FUNKION[SPACE]ENDE"
                  <158>
12060 PRINT X$;"BITTE[SPACE]WARTEN[SPACE]BIS
     [SPACE]";BR$:FF$(1)=MID$(B$(1),18,2)       <013>
12070 FF$(I)=MID$(B$(K),18,2)                   <097>
12080 FOR J=1 TO I-1                            <214>
12090 IF FF$(I)=FF$(J)THEN 12120                <216>

12100 NEXT                                      <245>
12110 I=I+1                                     <156>
12120 K=K+1:IF K>B THEN 12140                   <076>
12130 GOTO 12070                                <020>
12140 Z=I-1                                     <204>
12150 FOR C=1 TO Z-1                            <038>
12160 J=C+1:PRINT X$;"ELEMENT[SPACE7,LEFT4]";
      J-1;"[UP3]"                               <168>
12170 FOR D=J TO Z                              <120>
12180 IF FF$(C)>FF$(D)THEN 12200                <036>
12190 GOTO 12210                                <076>
12200 C$(0)=FF$(C):FF$(C)=FF$(D):FF$(D)=C$(0)
                  <067>
12210 NEXT                                      <099>
12220 NEXT                                      <109>
12230 PRINT X$;"BLOCKS[SPACE]WERDEN[SPACE]
     ADDIERT":BL$="[RVSON]DISK[SPACE]"
     :BM$="[RVSON]"+BF$                         <115>
12240 J=1                                       <044>
12250 FOR I=1 TO B:IF I=1 THEN GS=0             <176>
12260 IF MID$(B$(I),18,2)=FF$(J)THEN GOSUB 12350
                  <216>
12270 NEXT:FF$(J)=BF$+"[SPACE]DISK[SPACE]
     "+FF$(J)+"[SPACE]"+RIGHT$(ZK$+STR$(664-GS),3)
     :J=J+1                                     <105>
12280 IF J<Z+1 THEN 12250                       <133>
12290 PRINT X$;BR$                              <058>
12300 PRINT X$;"DRUCKER[SPACE]ODER[SPACE]SCREEN
     [SPACE](D/S)[SPACE]":GOSUB 11000:PRINT     <109>
12310 IF A$<>"S"AND A$<>"D"THEN GOSUB 11000
      :GOTO 12320                               <018>
12320 IF A$="D"THEN 12380                       <255>
12330 FOR I=1 TO Z:PRINT"[RIGHT4]";FF$(I)       <021>
12340 GOSUB 11000:NEXT:GOTO 12420               <091>
12350 G=0:G=VAL(RIGHT$(B$(I),3))                <089>
12360 GS=GS+G                                   <079>
12370 RETURN                                    <016>
12380 PRINT X$;"DRUCKER[SPACE]EINSCHALTEN[SPACE]
     ![SPACE]"                                  <127>
12390 GOSUB 10000                               <020>
12400 OPEN 4,4                                  <211>
12410 FOR I=1 TO Z:PRINT#4,"[SPACE5]";FF$(I)
      :NEXT:PRINT#4:CLOSE 4                     <032>
12420 PRINT V$;"DATEN[SPACE]AUF[SPACE]DISK
     [SPACE]SPEICHERN[SPACE]?[SPACE](J/N)"
     :GOSUB 11000:IF A$<>"J"THEN RETURN         <187>
12430 PRINT X$;"DATEN[SPACE]DISKETTE[SPACE]
     EINLEGEN[SPACE]"                           <098>
12440 GOSUB 10000:OPEN 2,8,2,FI$+",S,W"
      :PRINT#2,Z                                <108>
12450 FOR I=1 TO Z:PRINT#2,FF$(I):NEXT:CLOSE 2
                  <191>
12460 RETURN                                    <106>
14000 REM                                       <117>
14010 REM *** AUSSORTIEREN FREMDER LISTEN ***
                  <243>
14020 REM                                       <137>
14030 IF B<2 THEN RETURN                        <235>
14040 PRINT V$;"AUSSORTIEREN[SPACE]FREMDER
     [SPACE]LISTEN":DI$="@:DIFFERENZ":LI$="LISTE"
     :X=1                                       <239>
14050 PRINT X$;"MIT[SPACE]WELCHER[SPACE]LISTE
     [SPACE]VERGLEICHEN[SPACE]?":GOSUB 17000    <075>
14060 PRINT X$;"WIEVIEL[SPACE]ZEICHEN[SPACE]
     VERGLEICHEN[SPACE](1-20)[SPACE]";:GOSUB 18000
     :SK=VAL(K$)                                <024>
14070 IF SK>25 THEN SK=25                       <203>
14080 OPEN 5,8,5,FI$+",S,R":GOSUB 16000
      :IF T<>0 THEN 14100                       <162>
14090 OPEN 2,8,2,"@:LISTE"+",S,W":GOSUB 16000
                  <237>

14100 IF T<>0 THEN CLOSE 5:CLOSE 2:PRINT X$;K$
      :PRINT X$;ZM$:GOSUB 11000:RETURN          <021>
14110 INPUT#5,C                                 <125>
```

```
14120 FOR J=1 TO C                                  <244>
14130 INPUT#5,D$:PRINT"[RIGHT5]";D$                 <001>
14140 FOR I=1 TO B                                  <006>
14150 IF LEFT$(D$,SK)=LEFT$(B$(I),SK)THEN 14300
                                     <135>
14160 NEXT I                                        <082>
14170 PRINT#2,D$:X=X+1                              <102>
14180 NEXT J                                        <103>
14190 CLOSE 5:CLOSE 2:X=X-1                         <254>
14200 IF X=0 THEN PRINT"[CLEAR,DOWN,RVSON,SPACE]
   KEIN[SPACE]NEUES[SPACE]ELEMENT[SPACE]
   GEFUNDEN":GOTO 10000                             <095>
14210 OPEN 7,8,7,DI$+",S,W"                         <035>
14220 PRINT#7,X                                     <022>
14230 OPEN 5,8,5,LI$+",S,R"                         <054>
14240 FOR I=1 TO X:INPUT#5,D$:PRINT#7,D$:NEXT
                                     <096>
14250 CLOSE 7:CLOSE 5                               <199>
14260 PRINT V$;"DAS[SPACE]AUSSORTIEREN[SPACE]
   IST[SPACE]BEENDET[SPACE]"                        <224>
14270 PRINT X$;"DAS[SPACE]DATENFILE[SPACE]<
   [SPACE]DIFFERENZ[SPACE]>[SPACE]"                 <004>
14280 PRINT X$;"KANN[SPACE]GELADEN[SPACE]WERDEN
   [SPACE]"                                         <025>
14290 OPEN 15,8,15,"S0:LISTE":CLOSE 15
   :GOSUB 16000:RETURN                              <120>
14300 IF J=C THEN 14190                             <131>
14310 NEXT J                                        <233>
14320 RETURN                                        <181>
15000 REM                                           <097>
15010 REM *** ABFRAGE ROUTINEN ***                  <195>
15020 REM                                           <117>
16000 INPUT#3,T,K$,0,0:RETURN                       <078>
17000 PRINT X$;"FILE[SPACE]NAME[SPACE]";:F=1
   :GOSUB 18000:F=0:FI$=K$:RETURN                   <021>
18000 K$=""                                         <251>
18010 GOSUB 11000:IF A$=CHR$(13)THEN PRINT
   :RETURN                                          <127>
18020 IF A$=CHR$(20)AND LEN(K$)<>0 THEN K$=LEFT
   $(K$,LEN(K$)-1):PRINT A$;                        <218>
18030 IF DR=1 AND LEN(K$)=33 THEN 18010             <163>
18040 IF F=1 AND LEN(K$)=18 THEN 18010              <096>
18050 IF A$>CHR$(31)THEN K$=K$+A$:PRINT A$;
                                     <158>
18060 GOTO 18010                                    <085>
```

ESF: Editieren sequentieller Dateien

Oft kommt es vor, gerade beim Programmieren von Datei-Verwaltungen, daß sich Fehler einschleichen. Wenn sich diese Fehler innerhalb einer sequentiellen Datei befinden, hilft Ihnen dieses Programm, sie zu finden und zu verbessern.

Mit diesem Programm können Sie jede beliebige sequentielle Datei lesen, editieren, drucken und wieder neu abspeichern. Es gibt mehrere Anwendungsfälle für ESF:

1. Sie haben auf einer Diskette ein sequentielles File und möchten wissen, was dort gespeichert steht. Mit ESF ist das kein Problem.
2. Sie benutzen in einem Programm eine sequentielle Datei und erhalten bei der Ausgabe nicht das gewünschte Ergebnis. Um den Fehler schneller zu finden, können Sie mit ESF die Datei anschauen und drucken. So erkennen Sie zum Beispiel, ob schon beim Speichern Fehler gemacht wurden oder erst beim Lesen der Daten. Somit lassen sich Fehlerquellen eingrenzen. Auch Änderungen lassen sich einfach durchführen. Datensätze können gelöscht oder verändert, aber auch zusätzliche an beliebiger Stelle hinzugefügt werden.

Bedienung des Programms

ESF ist ein reines Basic-Programm und wird mit LOAD "ESF*",8 geladen. Es ist vollständig menügesteuert. Nach dem Start muß der Dateiname eingegeben werden. Sie sollten also vor dem Laden von ESF wissen, welche Datei bearbeitet werden soll. Die nächste Frage entscheidet über die Art des Einlesens, entweder mit GET# oder mit INPUT#.

Mit INPUT# wird jeder Datensatz bis zum nächsten Trennungszeichen (CHR$(13)) auf einmal gelesen. Mit der Leertaste werden weitere Sätze angezeigt. Jeder Datensatz wird numeriert (nur auf dem Bildschirm oder später auf dem Drucker, nicht in der Datei selbst). Die Nummern werden revers am linken Bildschirmrand ausgegeben. Sie erleichtern des Editieren.

Da Datensätze länger als 88 Zeichen sein können, kann auch GET# benutzt werden. Mit INPUT# gäbe es ein »STRING TOO LONG ERROR«. Dann wird mit jeder Betätigung der Leertaste ein einziges Zeichen gelesen. Diese Zeichen werden solange aneinandergefügt, bis das Trennungszeichen (CHR$(13)) erscheint. Es wird durch ein »*« sichtbar gemacht. Lesen mit GET# dauert zwar länger, jedoch können sämtliche Zeichen gelesen werden, also auch Kommata.

Wenn Sie einen Drucker haben, sollten Sie sich die Datei ausdrucken lassen. Das geht mit jedem Drucker, der mit OPEN 1,4 angesprochen werden kann. Auch hier werden, wie schon erwähnt, die Datensätze numeriert. Die Nummern brauchen sie beim Einfügen, Ändern und Löschen. Mit ihnen kann jeder Satz angesprochen werden.

Beim Einfügen und Ändern haben Sie ebenfalls die Möglichkeit, dies entweder mittels GET oder INPUT zu machen. GET sollte dann gewählt werden, wenn im Datensatz Kommata oder andere Zeichen vorkommen sollen, die INPUT nicht akzeptiert, oder bei Eingaben länger als 88 Zeichen. In jedem Fall wird die Eingabe mit der RETURN-Taste abgeschlossen.

Die Datei kann erneut abgespeichert werden, unabhängig, ob sie verändert wurde oder nicht. Sie kann sowohl einen neuen Namen erhalten (die bisherige Version wird nicht gelöscht), als auch überschrieben werden. Eine Sicherheitsabfrage verhindert ein versehentliches überschreiben. Anschließend startet ESF erneut.

(G. Kluge)

Variablenliste

TE$(I) = jedes Element enthält einen Datensatz
N$ = enthält den Dateinamen
FI$ = Dateiname plus Parameter zum Öffnen der Datei
A$ = eingelesenes Zeichen (GET) oder Zeile (INPUT)
ST = Statusvariable = 64, wenn Dateiende erreicht
A1,A2$,A3,A4 = Fehlermeldungen bei Floppy-Fehler
I = Anzahl der bisher gelesenen Datensätze
N = Nummer der Zeile, die geändert, eingefügt oder gelöscht werden soll.

```
100 REM ******************************         <223>
110 REM *        EDIT SEQ-FILE          *      <173>
120 REM *                               *      <091>
130 REM * EDITIEREN VON SEQUENTIELLEN*         <207>
140 REM *           DATEIEN             *      <105>
150 REM ******************************         <017>
160 :                                          <218>
170 :                                          <228>
180 DIM TE$(900):REM MAX ANZAHL DATENSATZ      <252>
190 OPEN 15,8,15                               <185>
200 PRINT"[CLEAR]"                             <056>
210 PRINT"*************************************
   ***"                                        <175>
220 PRINT"[DOWN,SPACE6]LESEN[SPACE]UND[SPACE]
   EDITIEREN"                                  <193>
230 PRINT"[DOWN,SPACE8]EINER[SPACE]SEQ.[SPACE]
   DATEI"                                      <197>
240 PRINT"[DOWN2,SPACE6]L[SPACE]=[SPACE]LESEN
   [SPACE]EINER[SPACE]DATEI"                   <201>
250 PRINT"[DOWN,SPACE6]B[SPACE]=[SPACE]
   PROGRAMMENDE[SPACE]"                        <232>
260 PRINT"*************************************
   ***"                                        <226>
270 PRINT "[DOWN,RVSON,SPACE]WAEHLE:[RVOFF]"
                    <145>
280 GET R$:IF R$="" THEN 280                   <162>
290 IF R$="B" THEN CLOSE 2:CLOSE 15:END        <207>
300 IF R$<>"L"THEN 280                         <099>
310 :                                          <113>
320 REM ---- LESEN EINER SEQ-FILE -----        <133>
330 :                                          <133>
340 :                                          <143>
350 I=1                                        <139>
360 PRINT"[DOWN2,SPACE]WELCHE[SPACE]DATEI[SPACE]
   SOLL[SPACE]GELESEN[SPACE]WERDEN?"           <200>
370 INPUT"DATEINAME[SPACE]=[SPACE]";N$         <174>
380 FI$=N$+",S,R"                              <063>
390 PRINT"*************************************
   ***"                                        <100>
400 PRINT"[DOWN2,SPACE2]SOLL[SPACE]MIT[SPACE]
   GET[SPACE]ODER[SPACE]MIT[SPACE]INPUT"       <056>
410 PRINT"[DOWN,SPACE2]GELESEN[SPACE]WERDEN
   [SPACE](G/I)[SPACE]?"                       <160>
420 GET RR$:IF RR$=""THEN 420                  <206>
430 IF RR$="G" THEN 480                        <130>
440 IF RR$="I" THEN 670                        <143>
450 GOTO 420                                   <226>
460 :                                          <007>
470 REM ---------- GET# ---------------        <206>
480 OPEN 2,8,2,FI$                             <083>
490 INPUT#15, A1,A2$,A3,A4                     <119>
500 IF A1<>0 THEN GOSUB 2320:GOTO 350          <220>
```

```
510 PRINT"[CLEAR]"                                   <111>
520 PRINT:PRINT"[SPACE8]LEERTASTE[SPACE]
   DRUECKEN[SPACE]!":PRINT                           <168>
530 PRINT"[SPACE8]---------------------":PRINT
                          <072>
540 PRINT"[RVSON]"I"[RVOFF]";                        <103>
550 GET R$:IF R$<>"[SPACE]"THEN 550                  <099>
560 GET#2,A$                                         <185>
570 IF A$=CHR$(13)THEN PRINT"*":I=I+1:GOTO 610
                          <187>
580 TE$(I)=TE$(I)+A$                                 <181>
590 PRINT  A$;                                       <137>
600 GOTO 550                                         <125>
610 IF ST<>64 THEN 540                               <164>
620 CLOSE 2                                          <064>

630 I=I-1                                            <152>
640 GOTO 800                                         <163>
650 :                                                <198>
660 :                                                <208>
670 REM -------INPUT# ----------------               <237>
680 OPEN 2,8,2,FI$                                   <028>
690 INPUT#15, A1,A2$,A3,A4                           <064>
700 IF A1<>0 THEN GOSUB 2320:GOTO 350                <165>
710 PRINT:PRINT"[SPACE8]LEERTASTE[SPACE]
   DRUECKEN[SPACE]!":PRINT                           <102>
720 GET R$:IF R$<>"[SPACE]"THEN 720                  <012>
730 INPUT#2,A$                                       <035>
740 TE$(I)=A$                                        <084>
750 PRINT "[RVSON]"I"[RVOFF]";A$                     <158>
760 IF ST<>64 THEN I=I+1:GOTO 720                    <028>
770 CLOSE 2                                          <215>
780 :                                                <073>
790 :                                                <083>
800 REM ------ EDITIEREN -------------               <130>
810 :                                                <103>
820 FLAG=0                                           <051>
830 PRINT"*************************************
   ***"                                              <030>
840 PRINT"[DOWN2,SPACE2]E[SPACE]=[SPACE]
   EINFUEGEN[SPACE]EINER[SPACE]ZEILE"                <078>
850 PRINT"[DOWN,SPACE2]C[SPACE]=[SPACE]AENDERN
   [SPACE3]EINER[SPACE]ZEILE"                        <172>
860 PRINT"[DOWN,SPACE2]L[SPACE]=[SPACE]LOESCHEN
   [SPACE2]EINER[SPACE]ZEILE"                        <019>
870 PRINT"[DOWN,SPACE2]D[SPACE]=[SPACE]DRUCKEN
   [SPACE]DER[SPACE]DATEI[SPACE4]"                   <038>
880 PRINT"[DOWN,SPACE2]2[SPACE]=[SPACE]WEITER
   [SPACE]"                                          <160>
890 PRINT"*************************************
   ***"                                              <090>
900 PRINT"[DOWN]WAEHLE"                              <043>
910 GET R$:IF R$=""THEN 910                          <026>
920 IF R$="E"THEN GOSUB 1470                         <215>
930 IF R$="C"THEN GOSUB 1030                         <215>
940 IF R$="L"THEN GOSUB 1620                         <239>
950 IF R$="D"THEN GOSUB 1730                         <243>
960 IF R$="2"THEN 1960                               <099>
970 GOTO 800                                         <238>
980 :                                                <017>
990 :                                                <027>
1000 REM ------- AENDERN EINER ZEILE --              <248>
1010 PRINT"************************************
   ****"                                             <210>
1020 :                                               <057>
1030 PRINT"[CLEAR]AENDERN[SPACE]EINER[SPACE]
   ZEILE"                                            <099>
1040 PRINT"[DOWN,SPACE]I=[SPACE]MIT[SPACE]INPUT
   [SPACE](KEIN[SPACE]KOMMA[SPACE]!)"                <016>
1050 PRINT"[DOWN,SPACE]G=[SPACE]MIT[SPACE]GET
   [SPACE](SCHREIBFEHLER[SPACE]VERMEIDEN!)"
                          <033>
1060 PRINT"[DOWN,SPACE]2=[SPACE]WEITER"              <085>
1070 PRINT"************************************
   ****"                                             <015>
1080 PRINT"[DOWN2]WAEHLE"                            <241>
1090 GET R$:IF R$=""THEN 1090                        <255>
1100 IF R$="2"THEN RETURN                            <174>
1110 IF R$="G"THEN 1300                              <003>
1120 IF R$="I"THEN 1140                              <017>
1130 GOTO 1030                                       <187>
1140 REM --------INPUT ---------------               <162>
1150 IF FLAG=1 THEN 1220                             <118>
1160 PRINT"************************************
   ****"                                             <105>
1170 PRINT"WELCHE[SPACE]ZEILE[SPACE]SOLL[SPACE]
   GEAENDERT[SPACE]WERDEN?"                          <113>
```

```
1180 INPUT"NR:";N$:N=VAL(N$)                         <178>
1190 IF N<1 OR N>I THEN PRINT
   :PRINT TAB(8)"NICHT[SPACE]MOEGLICH![SPACE]"
   :PRINT:PRINT:GOTO 1160                            <202>
1200 PRINT"ALTER[SPACE]INHALT
   :[SPACE,RVSON]"TE$(N)                             <113>
1210 PRINT                                           <087>
1220 PRINT"[DOWN,SPACE]GEBEN[SPACE]SIE[SPACE]
   JETZT[SPACE]DEN[SPACE]NEUEN[SPACE]INHALT
   [SPACE]EIN[SPACE](INPUT)[SPACE]"                  <088>
1230 INPUT A$                                        <188>
1240 TE$(N)=A$                                       <079>
1250 PRINT"[DOWN,RVSON]"TE$(N)                       <066>
1260 FLAG=0                                          <236>
1270 RETURN                                          <136>
1280 REM ----MIT GET ---------------                 <226>
1290 PRINT"************************************
   ****"                                             <236>
1300 IF FLAG=1 THEN 1330                             <015>
1310 PRINT"WELCHE[SPACE]ZEILE[SPACE]SOLL[SPACE]
   GEAENDERT[SPACE]WERDEN?"                          <254>
1320 INPUT"NR:";N$:N=VAL(N$)                         <063>
1330 PRINT"ALTER[SPACE]INHALT[SPACE]";TE$(N)
   :TE$(N)=""                                        <111>
1340 PRINT"************************************
   ****"                                             <030>
1350 PRINT"[DOWN,SPACE]GEBEN[SPACE]SIE[SPACE]
   JETZT[SPACE]DEN[SPACE]NEUEN[SPACE]INHALT
   [SPACE]EIN[SPACE](GET):[SPACE2]":PRINT            <056>
1360 GET A$:IF A$="" THEN 1360                       <236>
1370 IF A$=CHR$(13) THEN 1420                        <235>
1380 PRINT A$;                                       <162>
1390 TE$(N)=TE$(N)+A$                                <236>
1400 GOTO 1360                                       <208>
1410 :                                               <193>
1420 :                                               <203>
1430 PRINT"[DOWN,RVSON]"TE$(N)                       <247>
1440 FLAG=0                                          <161>
1450 RETURN                                          <061>
1460 :                                               <243>
1470 REM ------EINFUEGEN EINER ZEILE--               <060>
1480 :                                               <007>
1490 PRINT"************************************
   ****"                                             <180>
1500 PRINT"HINTER[SPACE]WELCHER[SPACE]ZEILE"
                          <011>
1510 INPUT N                                         <190>
1520 N=N+1                                           <030>
1530 FOR L=I TO N STEP-1                             <062>
1540 :TE$(L+1)=TE$(L)                                <133>
1550 NEXT L                                          <226>
1560 I=I+1                                           <061>
1570 FLAG=1                                          <037>
1580 PRINT"SCHREIBEN[SPACE]DER[SPACE]NEUEN
   [SPACE]ZEILE[SPACE]"                              <113>
1590 GOTO 1040                                       <138>
1600 :                                               <128>
1610 :                                               <138>
1620 REM-------LOESCHEN---------------               <024>
1630 :                                               <158>
1640 PRINT:PRINT                                     <218>
1650 PRINT"************************************
   ****"                                             <085>
1660 INPUT"[SPACE]WELCHE[SPACE]ZEILE[SPACE]
   LOESCHEN?";N                                      <149>
1670 IF N>I THEN PRINT"MAX=[SPACE]"I:GOTO 1710
                          <219>
1680 FOR L=N TO I:TE$(L)=TE$(L+1):NEXT               <135>
1690 I=I-1                                           <192>
1700 PRINT"ZEILE"N"[SPACE]IST[SPACE]GELOESCHT"
                          <032>
1710 RETURN                                          <066>
1720 :                                               <248>

1730 REM ------ DRUCKEN DER DATEI ------             <193>
1740 PRINT                                           <107>
1750 PRINT"************************************
   ****":PRINT                                       <140>
1760 PRINT"DRUCKEN[SPACE]DER[SPACE]DATEI[SPACE]"
                          <017>
1770 PRINT"[DOWN,SPACE]1=[SPACE]DRUCKEN"             <088>
1780 PRINT"[DOWN,SPACE]2=[SPACE]WEITER"              <039>
1790 PRINT"[SPACE2]WAEHLE"                           <151>
1800 GET R$:IF R$=""THEN 1800                        <199>
1810 IF R$<"1"OR R$>"2"THEN 1800                     <255>
1820 IF R$="2"THEN 1910                              <190>
```

```
1830 INPUT"DRUCKER[SPACE]OK?";R$                    <144>
1840 OPEN 1,4:CMD 1                                 <111>
1850 PRINT"DATEI:[SPACE]"N$                         <049>
1860 FOR J=1 TO I                                   <230>
1870 :PRINT J;TE$(J)                                <005>
1880 NEXT J                                         <043>
1890 PRINT:PRINT:PRINT:PRINT                        <123>
1900 PRINT#1:CLOSE 1                                <071>
1910 RETURN                                         <011>
1920 :                                              <193>
1930 :                                              <203>
1940 REM ----- SPEICHERN DER DATEI ----             <162>
1950 PRINT"************************************
   ****"                                            <130>
1960 PRINT"SOLL[SPACE]DER[SPACE]GESAMTE[SPACE]
   TEXT[SPACE]ERNEUT[SPACE]ABGESPEICHERT[SPACE]
   WERDEN"                                          <058>
1970 PRINT"[DOWN,SPACE]1=[SPACE]JA"                 <160>
1980 PRINT"[DOWN,SPACE]2=[SPACE]PROGRAMMENDE"
                         <161>
1990 PRINT"************************************
   ****"                                            <170>
2000 PRINT"[DOWN,RVSON]WAEHLE"                      <141>
2010 GET R$:IF R$=""THEN 2010                       <147>
2020 IF R$="2" THEN RUN                             <069>
2030 IF R$<>"1"THEN 2010                            <057>
2040 PRINT                                          <152>
2050 PRINT"WELCHER[SPACE]DATEINAME"                 <121>
2060 INPUT N$                                       <011>
2070 FI$=N$+",S,W"                                  <229>
2080 PRINT"[CLEAR]"                                 <120>
2090 CLOSE 2:OPEN 2,8,2,FI$                         <176>
2100 INPUT#15, A1,A2$,A3,A4                         <200>
2110 IF A1=0 THEN 2200                              <144>
2120 IF A1<>63 THEN GOSUB 2320:GOTO 1960            <168>
2130 PRINT:PRINT"[SPACE]DER[SPACE]NAME[SPACE]
   "N$"[SPACE]EXISTIERT[SPACE]SCHON!"               <025>
2140 PRINT:PRINT"[SPACE]WOLLEN[SPACE]SIE[SPACE]
   DIESE[SPACE]DATEI[SPACE]UEBERSCHREIBEN?"
                         <220>
2150 PRINT:PRINT"[SPACE]WAEHLE[SPACE](J/N)
   [SPACE]"                                         <236>
2160 GET R$:IF R$=""THEN 2160                       <048>
2170 IF R$="J"THEN N$="@:"+N$:GOTO 2070             <244>
2180 IF R$<>"N"THEN 2160                            <243>
2190 PRINT:PRINT"[SPACE]VERSUCHEN[SPACE]SIE
   [SPACE]ES[SPACE]NOCH[SPACE]EINMAL":GOTO 1960
                         <227>
2200 FOR J=1 TO I                                   <059>
2210 :  PRINT#2,TE$(J) :PRINT"[HOME,DOWN]"J,I
                         <044>
2220 NEXT J                                         <128>
2230 :                                              <248>
2240 PRINT"GESPEICHERT!"                            <249>
2250 CLOSE 2                                        <164>
2260 FOR I=1 TO 1500:NEXT                           <175>
2270 RUN                                            <112>
2280 :                                              <042>
2290 :                                              <052>
2300 PRINT"[CLEAR]"                                 <116>
2310 PRINT"************************************
   ****[DOWN2]"                                     <014>
2320 PRINT"************************************
   ****[DOWN2]"                                     <024>
2330 PRINT"[SPACE13]DISKETTENFEHLER![DOWN2]"
                         <164>
2340 PRINT"[SPACE,RVSON,SPACE]DATEI[SPACE]
   :[SPACE]"N$                                      <047>
2350 PRINT"[RVOFF]":REM REVERS OFF                  <033>
2360 PRINT A1,A2$,A3;A4                             <095>
2370 CLOSE 2                                        <029>
2380 PRINT"[DOWN2,SPACE7]BEHEBEN[SPACE]SIE
   [SPACE]DEN[SPACE]FEHLER[SPACE]UND[SPACE4]"
                         <146>
2390 PRINT"[DOWN2,SPACE2]........DRUECKEN[SPACE]
   SIE[SPACE]>F<[SPACE].........."                  <140>
2400 GET R$:IF R$<>"F" THEN 2400                    <022>
2410 PRINT"[CLEAR]"                                 <227>
2420 RETURN                                         <011>
```

Track 18 — Das Chaos organisieren

Nur der Dumme räumt auf, das Genie beherrscht das Chaos. Aber mal ganz ernsthaft, wem ist es noch nicht passiert, daß nach einer Kopieraktion von drei zusammengehörigen Programmen sich zwei davon auf der falschen Diskette wiederfanden, oder daß ihm die Reihenfolge seiner Programme nicht gefiel und er bei dem Versuch das zu ändern nach einigen Stunden SCRATCH & SAVE verzweifelt aufgab?

Dies muß ein Ende haben, dachte ich mir und ging daran, ein Werkzeug zu entwickeln, daß das Chaos beherrscht und Ordnung in wirre Diskettenverhältnisse bringt. Ob das Programm deswegen geniale Züge trägt, verbietet mir die Bescheidenheit zu diskutieren. Bleiben wir bei den Fakten.

Track 18 ist von der Grundkonzeption her dafür gedacht, ein Diskettendirectory umzusortieren und an geeigneter Stelle Bemerkungen anzubringen. Darüber hinaus bietet es einige nützliche und teilweise nicht alltägliche Features. So zum Beispiel die Möglichkeit, eine Bemerkung (etwa Informationen über einzelne Programme) in einen freien Bereich auf Spur 18 zu schreiben. Sollte jemand, wie ich auch, es vorziehen, Schreibschutzaufkleber zum Beschriften seiner Disketten zu verwenden, es sei ihm gewährt - Track 18 bietet auch einen Softschutz. Auch die Möglichkeit einzelne Files zu schützen, die das DOS der 1541 vorsieht, wird genutzt.

Das Programm arbeitet ausschließlich mit Spur 18 (Hex $12), wenn man von den Auswirkungen einiger Diskettenbefehle absieht.

Die Programmbedienung wird dem Anwender durch konsequente Menüführung leicht gemacht.

Der Aufbau en Detail:

Das Hauptmenü:

Directory einlesen
Directory bearbeiten
Neues Directory schreiben
Directory von Disk

Disk Command senden
Diskettennamen ändern
Disk-Kommentar
Files schützen/freigeben
Schreibschutz

Ende

Der aktuelle Menüpunkt ist durch reverse Schrift hervorge-

hoben. Die Auswahl der Positionen erfolgt über die Cursor-up/down Taste, die den reversen »Balken« verschiebt. Aus dem angewählten Programmteil kann zu jedem Zeitpunkt durch F1 ins Hauptmenü zurückgekehrt werden.

Directory einlesen

Dieser Menüpunkt gehört zu dem auch optisch abgetrennten Komplex »Directory bearbeiten«. Das Directory muß zur Bearbeitung in den Speicher gelesen werden. Nur hier wird umgestellt, sortiert und eingefügt. Und um genau diese Dinge zu tun, bewege der geneigte Anwender den reversen Balken auf den Menüpunkt.

Directory bearbeiten

Sofort nach Betätigen der Return-Taste (vorausgesetzt man hat vorher das Directory eingelesen) erscheint ein komplettes Inhaltsverzeichnis auf dem Bildschirm. Die bis zu 144 Files werden auf 9 Seiten dargestellt, die mit der Cursor-up/down Taste umgeblättert werden können. Komplett heißt, auch freie Plätze, die durch SCRATCH entstanden sind, werden angezeigt und können mit den Funktionen

F3 = Sortieren
F5 = Einfügen
F7 = Löschen

wie jeder andere Fileeintrag behandelt werden. So kann ein Freiplatz an eine Stelle verschoben werden, wo beim nächsten SAVE der Programmname aufgenommen werden soll.

Sortieren

Die Routine ermöglicht das Verschieben eines Files von einer Position zu einer anderen. Nach Drücken von F3 wird erst nach der Platznummer des Files gefragt, das verschoben werden soll, dann wohin es geschoben werden soll.

Einfügen

Hier wird nach der Platznummer gefragt, vor die eine Bemerkung gestellt werden soll. Da das System niemandem vorschreibt, welcher Art eine solche Bemerkung sein soll, — es wird unverbindlich eine Reihe Bindestriche angeboten — fragt es noch nach dem Bemerkungstext. Dieser wird dann als Programmfile getarnt in die Inhaltsliste aufgenommen. Ein Versuch diesen »blinden Passagier« durch LOAD später aufzurufen endet allerdings mit einem ERROR, da der Verweis auf den Fileanfang auf der Diskette fehlt.

Löschen

Dieser Menüpunkt ist mit etwas Vorsicht zu genießen. Nach Angabe der gewünschten Platznummer wird dieser Platz, ob Bemerkung, ob File, schonungslos gestrichen — wohlgemerkt nur im Speicher. Erst die Rückkehr ins Hauptmenü und dann die Anwahl des Punktes.

Neues Directory schreiben

bannt alle Veränderungen auf die Diskette. Da dieser Programmteil alle, auch ungewollte Änderungen am Directory endgültig macht, wird zur Sicherheit noch einmal eine Befehlsbestätigung verlangt. Sollte man sich seiner Sache nicht ganz sicher sein, ist es ratsam, noch einmal einzulesen.

Ein auf die Diskette zurückgeschriebenes Inhaltsverzeichnis ist natürlich ohne Einschränkungen funktionsfähig.

Directory von Disk

Dieser Programmteil stellt das im Moment tatsächlich auf der Diskette vorhandene Directory dar, ebenfalls in Seiten zu je 16 Files. Die Routine ist nicht übermäßig schnell und kann daher zu jedem Zeitpunkt mit F1 abgebrochen werden. Man kann auf diese Weise sofort den Erfolg seiner Arbeit kontrollieren, denn die Art der Darstellung entspricht genau dem »LOAD "$"« Befehl in Basic.

Die im Folgenden beschriebenen Programmfunktionen beinhalten praktisch alles, was man braucht, wenn man die in Spur 18 enthaltenen Informationen verändert. Zunächst jedoch die einzige Routine, die etwas aus dem Rahmen fällt, weil sie nicht auf Spur 18 begrenzt ist:

Disk-Kommando senden

Ohne OPEN-Anweisung kann jeder Befehl, den das DOS versteht, an die Diskettenstation geschickt werden. Das Format der Befehle ist dasselbe wie bei »OPEN«. Jedoch wird nur der Teil gebraucht, der normalerweise in Anführungszeichen steht. Ein Befehl darf bis zu 70 Zeichen lang sein.

Diskettennamen ändern

Wer möchte nicht manchmal eine Diskette umbenennen, ohne sie gleich zu formatieren? Name und ID sind gemeinsam oder einzeln veränderbar, ohne die übrigen Informationen auf der Diskette zu berühren. Ein echter Leckerbissen.

Disk-Kommentar

Hier wird brachliegender Speicherplatz sinnvoll genutzt! In 80 freie Bytes des Block 0 auf Spur 18 kann eine 80 Zeichen lange Bemerkung geschrieben werden. Nach dem Anwählen dieses Menüpunktes gelangt man in ein Untermenü mit den Positionen:

Lesen

hier wird nur der vorhandene Text ausgelesen.

Schreiben

Vor jedem Schreibvorgang wird der schon vorhandene Text eingelesen und im vier Zeilen langen Arbeitsfeld dargestellt. Er kann nun überschrieben oder zeilenweise editiert werden.

Files schützen/freigeben

Das DOS bietet eine Möglichkeit, einzelne Files vor versehentlichem »exitus« zu beschützen. Ein geschütztes File kann nicht geSCRATCHt werden. Es ist höchstens noch mit der Löschen-Funktion von »Track 18« auszuradieren (außer man gibt es vorher wieder frei).

Und hier ist die vielleicht wichtigste Routine:

Schreibschutz

Was bisher nur mit einem Klebeschild zu bewerkstelligen war, findet nun unsichtbar softmäßig auf der Diskette statt. Eine Diskette, die mit »Track 18« geschützt ist, verweigert jeglichen Schreibzugriff. Lesen ist weiterhin problemlos möglich. Als Kennzeichen steht in der Namenszeile des Directory beim Laden nicht mehr 2A, sondern zwei »Kleiner«-Zeichen.

Der Vollständigkeit halber befindet sich noch die Anweisung

Ende

im Menü. Das Programm beendet sich nach einer Rückfrage selbst mit einem normalen END. Wer möchte, kann statt der END-Anweisung den Befehl SYS 64738 einfügen. Damit findet dann ein Reset statt, und der Computer steht für weitere Aufgaben wieder jungfräulich zur Verfügung.

Nach dieser ausführlichen Erläuterung bleibt mir nur noch ein Ratschlag: Für Probeläufe nach dem Eintippen und zum Einarbeiten auf jeden Fall eine besondere Probediskette verwenden, damit bei Eingabefehlern keine wichtigen Daten oder Programme zerstört werden.

Tips zum Tippen

Die im Programm verwendeten REM-Zeilen dienen nur der optischen Gliederung des Listings. Sie werden von keiner Sprunganweisung angesprungen, so daß man ohne unangenehme Folgen darauf verzichten kann.

Die folgende »PRINT AT«-Maschinenroutine habe ich dem

Data Becker Buch »64 Intern« entnommen. Sie ist die kürzeste, die mir bekannt ist. Im Programm wird sie mit SYS P,x,y aufgerufen, wobei P=828 (=$033C).

```
033C JSR  $AEFD   prüft auf Komma
033F JSR  $B79E   holt numerischen Wert ins X-Register
0342 TXA
0343 PHA
0344 JSR  $AEFD
0347 JSR  $AEFD
034A PLA
034B TAY
034C CLC
034D JMP  $FFF0   Cursor setzen
```

Allen leidenschaftlichen Programmierern und Programmverbesserern sei gesagt: »Track 18« ist ausbaufähig! Für diesen Zweck habe ich mir Mühe gegeben, das Programm modular aufzubauen und zumindest die wichtigsten Routinen wasserdicht zu machen. Das heißt, vor dem Aufruf mit GOSUB werden die benötigten Werte in einige festgelegte Variablen geschrieben und die Routine gibt ihre Antwort über andere festgelegte Variablen. Jede andere von der Routine benutzte Variable wird direkt am Ort des Geschehens neu belegt (Arrays und Programmkonstanten ausgenommen). Wer diesem Prinzip treu bleibt, wird bei einem Ausbau des Programms keine Probleme haben.

(Andreas Kölbach / ev)

```
0 REM :::::::::::::::::::::::::::::::::::    <009>
1 REM ::  TRACK 18  MANIPULATIONEN    ::     <122>
2 REM ::          - BY MATAN -        ::     <223>
3 REM ::                              ::     <122>
4 REM ::  VON     ANDREAS KOELBACH    ::     <165>
5 REM ::          MARBURG (1984)      ::     <179>
6 REM ::          STADTWALDSTR. 5     ::     <129>
7 REM :::::::::::::::::::::::::::::::::::    <016>
8 REM                                        <151>
9 REM*********** INITIAL **************        <188>
10 DATA 32,174,"",253,32,"SEQ",174,158,"PRG",32,
   183,"USR",158,104,"REL",183,168           <249>
11 DATA"",138,24,"SEQ<",72,76,"PRG<",32,240,
   "USR<",253,255,"REL<"                     <154>
15 POKE 53280,6:POKE 53281,6:POKE 646,1      <014>
20 DIM N$(145),C$(10)                        <060>
21 U=1:P=828:O$=CHR$(0):SP$="[SPACE20]"
   :REM 20 SPC                               <218>
22 FR$=O$+O$+O$+"-----[SPACE]FREI[SPACE]
   -----"+O$                                 <036>
23 FOR I=0 TO 9:FR$=FR$+O$:READ X,Y,TY$(I)
   :POKE 828+I,X:POKE 838+I,Y:NEXT I         <049>
25 GOTO 50000                                <151>
34 REM****** EINFACHES GET ************      <011>
35 POKE 198,0:WAIT 198,1:GET A$:A=ASC(A$):RETURN
                     <052>
49 REM********* GETBILD ***************      <171>
50 B$="":X=0                                 <002>
51 SYS P,CX,CY                               <176>
52 POKE 207,0:REM BLINKPHASE AUS             <083>
53 POKE 204,0:REM CURSOR EIN                 <065>
54 POKE 198,0:WAIT 198,1:GET A$:A=ASC(A$)    <127>
55 POKE 204,1:REM CURSOR AUS                 <081>
56 B=1024+PEEK(211)+PEEK(214)*40
   :POKE B,PEEK(B)AND 127                    <133>
57 IF A=20 AND X>0 THEN CX=CX+(X<L)
   :POKE B+(X<L),BC:X=X-1:B$=LEFT$(B$,X):GOTO 51
                     <238>
58 IF A=13 THEN A=1:RETURN                   <175>
59 IF A=133 THEN A=2:RETURN                  <228>
60 IF A<G OR A>H THEN 52                     <250>
61 IF X=L THEN B$=LEFT$(B$,L-1)              <176>
62 PRINT A$:B$=B$+A$:X=LEN(B$)               <193>
63 CX=CX-(X<L):GOTO 51                       <163>
99 REM****** INHALT SEITENWEISE *******       <025>
100 S=0                                      <153>
110 GOSUB 22000:PRINT"[SPACE]NR."TAB(5);"[RVSON]
   "D$"[RVOFF]"                              <164>
120 C=S*16+16*(S>9):X=0                      <146>
130 C=C+1:IF C>Q THEN RETURN                 <212>
140 X=X+1                                    <201>
150 BL=ASC(RIGHT$(N$(C),2))+ASC(RIGHT$(N$(C),
   1))*256                                   <242>
160 A=ASC(LEFT$(N$(C),1)):A$=TY$((A AND 7)-5*((
   A AND 64)>0))                             <122>
170 PRINT C;TAB(4)":";BL;TAB(10);MID$(N$(C),4,
   16);TAB(28);A$                            <208>
180 IF X<16 THEN 130                         <236>
190 RETURN                                   <076>
199 REM***** DISKNAMEN EINLESEN *******       <059>
200 GOSUB 260                                <237>
205 OPEN 1,8,15,"I":OPEN 2,8,2,"#"           <172>
210 PRINT#1,"U1[SPACE]2[SPACE]0[SPACE]18[SPACE]
   0"                                        <140>
220 PRINT#1,"B-P[SPACE]2[SPACE]144"          <159>
230 DN$=""                                   <146>
240 FOR I=1 TO 23:GET#2,A$:DN$=DN$+A$:NEXT I
                     <211>
250 RETURN                                   <136>
259 REM******** CHECK DISK ************       <100>
260 OPEN 1,8,15:CLOSE 1                      <214>
265 IF ST<>0 THEN 280:REM ANGESCHALTET ?     <140>
270 OPEN 1,8,15,"I":INPUT#1,A                <234>
275 IF A=0 THEN CLOSE 1:RETURN
   :REM DISK EINGELEGT ?                     <201>
280 POKE 53281,2:PRINT"[CLEAR]":SYS P,13,12
   :PRINT"LAUFWERK[SPACE]OK[SPACE]?"         <083>
285 GOSUB 35:POKE 53281,6:GOTO 50000         <001>
299 REM***** DIRECTORY EINLESEN *******       <187>
300 PRINT"[CLEAR,SPACE8]-[SPACE]DIRECTORY
   [SPACE2]EINLESEN[SPACE]-"                 <255>
320 GOSUB 200:D$=DN$:SYS P,8,4
   :PRINT"[RVSON]"D$"[RVOFF]"                <189>
330 S=1:C=0                                  <224>
340 PRINT#1,"U1[SPACE]2[SPACE]0[SPACE]18";S
                     <109>
350 GET#2,T$:GET#2,S$:T=ASC(T$+O$):S=ASC(S$)
                     <133>
360 FOR I=0 TO 7:C=C+1:N$(C)=""              <115>
370 PRINT#1,"B-P[SPACE]2";2+I*32             <014>
380 GET#2,A$                                 <004>
390 IF A$=""THEN N$(C)=FR$:GOTO 420          <225>
400 N$(C)=A$                                 <174>
410 FOR J=0 TO 28:GET#2,A$:N$(C)=N$(C)+LEFT$(A$
   +O$,1):NEXT J                             <183>
420 SYS P,0,8:PRINT"NR."C                    <123>
430 SYS P,9,8:PRINT MID$(N$(C),4,16)         <077>
440 NEXT I                                   <132>
450 IF T THEN 340                            <224>
460 Q=C:GOTO 50000                           <203>
599 REM****** FILE SCHUETZEN **********       <097>
600 P$="[CLEAR,SPACE8]-[SPACE]FILES[SPACE2]
   SCHUETZEN[SPACE]-"                        <221>
620 C$(3)="FILE[SPACE]SCHUETZEN[RVOFF]"      <186>
630 C$(4)="FILE[SPACE]FREIGEBEN[RVOFF]"      <147>
640 CX=11:CY=11                              <027>
650 GOSUB 18000                              <018>
670 CX=0:CY=15:SYS P,CX,CY:PRINT"NAME[SPACE]DES
   [SPACE]FILES[SPACE]:[SPACE]................"
                     <099>
680 CX=17:L=16:G=32:H=127:BC=46:GOSUB 50     <107>
690 ON A GOTO 700,50000                      <199>
700 L=LEN(B$)                                <054>
710 T=18:S=1:X=-1:GOSUB 260                  <038>
720 OPEN 1,8,15:OPEN 2,8,2,"#"               <248>
730 T1=T:S1=S                                <042>
740 PRINT#1,"U1[SPACE]2[SPACE]0[SPACE]18";S1
                     <047>
750 GET#2,T$:GET#2,S$:T=ASC(T$+O$):S=ASC(S$)
                     <022>
760 FOR I=0 TO 7:C$=""                       <024>
770 PRINT#1,"B-P[SPACE]2";2+32*I             <160>
780 GET#2,A$:IF A$=""THEN 810                <246>
790 PRINT#1,"B-P[SPACE]2";5+32*I:FOR J=1 TO L
   :GET#2,A$:C$=C$+A$:NEXT J                 <031>
800 IF C$=B$THEN X=I:I=7                     <115>
810 NEXT I                                   <248>
820 IF T=0 AND X<0 THEN P$="FILE[SPACE]NICHT
   [SPACE]GEFUNDEN[SPACE]!!":GOTO 930        <118>
830 IF X<0 THEN 730                          <072>
840 P$="-[SPACE]OK[SPACE]-"                  <169>
850 PRINT#1,"U1[SPACE]2[SPACE]0[SPACE]18";S1
                     <158>
860 PRINT#1,"B-P[SPACE]2";X*32+2             <009>
870 GET#2,A$:A=ASC(A$)                       <153>
880 IF F=1 THEN A=A OR 64                    <028>
890 IF F=2 THEN A=A AND 191                  <087>
```

```
900 PRINT#1,"B-P[SPACE]2";X*32+2                    <049>
910 PRINT#2,CHR$(A);                                <027>

920 PRINT#1,"U2[SPACE]2[SPACE]0[SPACE]18";S1
                              <229>
930 SYS P,0,18:PRINT P$:GOSUB 35:GOTO 50000
                              <178>
999 REM**** DIR. ZURUECKSCHREIBEN *****            <188>
1000 PRINT"[CLEAR,SPACE7]-[SPACE]DIRECTORY
   [SPACE]SCHREIBEN[SPACE]-"                       <253>
1010 GOSUB 260                                     <026>
1020 SYS P,10,12:PRINT"SIND[SPACE]SIE[SPACE]
   SICHER[SPACE]?"                                 <044>
1030 CY=15:GOSUB 20000                             <119>
1040 ON F GOTO 1050,50000                          <091>
1050 IF Q=0 THEN SYS P,5,12:PRINT"[RVSON]KEINE
   [SPACE]FILES[SPACE]EINGELESEN[SPACE]!!!
   [RVOFF]":GOSUB 35:GOTO 50000                    <036>
1060 C=0:T=18:S=1:A=T:B=S+3:A$=LEFT$(FR$,2)
                              <201>
1070 OPEN 1,8,15                                   <248>
1080 OPEN 2,8,2,"#"                                <098>
1090 PRINT#1,"B-P[SPACE]2[SPACE]0"                 <160>
1100 IF C*8+8>=Q THEN A=0:B=255                    <044>
1110 PRINT#2,CHR$(A);CHR$(B);                      <121>
1120 FOR I=1 TO 7:J=C*8+I                          <060>
1130 PRINT#2,N$(J);A$;                             <076>
1140 NEXT I                                        <067>
1150 PRINT#2,N$(J+1);                              <155>
1160 PRINT#1,"U2[SPACE]2[SPACE]0";T;S              <203>
1170 C=C+1:A=T:S=S+3:B=S+3                         <247>
1180 IF S=21 THEN 1220                             <255>
1190 IF S>18 THEN S=S-17                           <180>
1200 IF B>18 THEN B=B-17                           <139>
1210 IF C*8<Q THEN 1090                            <229>
1220 GOTO 50000                                    <070>
1499 REM****** DISKNAMEN AENDERN ******            <254>
1500 PRINT"[CLEAR,SPACE4]-[SPACE]DISKETTENNAMEN
   [SPACE2]AENDERN[SPACE]-"                        <194>
1510 GOSUB 200:SYS P,6,5:PRINT"[RVSON]"DN$"
   [RVOFF]"                                        <170>
1520 SYS P,0,10:PRINT"NAME:[SPACE]
   ................"                               <030>
1530 PRINT"[DOWN,SPACE2]ID:[SPACE].."              <016>
1540 SYS P,12,22:PRINT"F1[SPACE]=[SPACE]MENUE"
                              <092>
1550 CX=6:CY=10:L=16:BC=46:G=32:H=127:GOSUB 50
   :ON A GOTO 1560,1650                            <229>
1560 IF B$=""THEN DN$=LEFT$(DN$,LEN(DN$)-5)
   :GOTO 1580                                      <074>
1570 DN$=B$:FOR I=1 TO 18-LEN(DN$)
   :DN$=DN$+CHR$(160):NEXT                         <204>
1580 CX=6:CY=12:L=2:GOSUB 50                       <054>
1590 ON A GOTO 1600,1650                           <086>
1600 DN$=DN$+B$                                    <116>
1610 PRINT#1,"U1[SPACE]2[SPACE]0[SPACE]18[SPACE]
   0"                                              <010>
1620 PRINT#1,"B-P[SPACE]2[SPACE]144"               <029>
1630 PRINT#2,DN$;                                  <075>
1640 PRINT#1,"U2[SPACE]2[SPACE]0[SPACE]18[SPACE]
   0"                                              <041>
1650 GOTO 50000                                    <246>
1999 REM***** KOMMENTAR AUF DISK ******            <232>
2000 PRINT"[CLEAR,SPACE5]-[SPACE]KOMMENTAR
   [SPACE]AUF[SPACE]DISKETTE[SPACE]-"              <136>
2010 C$(3)="LESEN[RVOFF]"                          <203>
2020 C$(4)="SCHREIBEN[RVOFF]"                      <242>
2030 CX=15:CY=13                                   <148>
2040 GOSUB 18000                                   <133>
2050 SYS P,7,10:PRINT"**********************"
                              <127>
2060 SYS P,7,20:PRINT"**********************"
                              <138>
2070 GOSUB 260:OPEN 1,8,15:OPEN 2,8,2,"#"
   :REM LESEN                                      <227>
2080 PRINT#1,"U1[SPACE]2[SPACE]0[SPACE]18[SPACE]
   0"                                              <194>
2090 PRINT#1,"B-P[SPACE]2[SPACE]175"               <249>
2100 FOR I=1 TO 4:C$(I)=""                         <242>
2110 FOR J=1 TO 20:GET#2,A$:C$(I)=C$(I)+LEFT$(A
   $+"[SPACE]",1):NEXT J,I                         <151>
2120 IF F=2 THEN 2140                              <115>
2130 CY=12:FOR I=1 TO 4:SYS P,8,CY:PRINT C$(I)
   :CY=CY+2:NEXT:GOTO 2270                         <060>
2140 SYS P,0,12:REM SCHREIBEN                      <153>
2150 FOR I=1 TO 4:PRINT"[SPACE7]>"C$(I)"<[DOWN]"
   :NEXT                                           <150>
2160 CX=8:CY=12:J=1                                <079>
2170 L=20:BC=32:G=32:H=127:GOSUB 50                <074>
2180 ON A GOTO 2190,2280                           <171>
2190 IF LEN(B$)THEN B$=B$+LEFT$(SP$,20-LEN(B$))
   :C$(J)=B$                                       <081>
2200 CX=8:CY=CY+2:J=J+1                            <128>
2210 IF J<5 THEN 2170                              <216>
2230 PRINT#1,"U1[SPACE]2[SPACE]0[SPACE]18[SPACE]
   0"                                              <120>
2240 PRINT#1,"B-P[SPACE]2[SPACE]175"               <143>
2250 PRINT#2,C$(1);C$(2);C$(3);C$(4);              <094>
2260 PRINT#1,"U2[SPACE]2[SPACE]0[SPACE]18[SPACE]
   0"                                              <151>
2270 SYS P,8,21:PRINT"-[SPACE]OK[SPACE]-"
   :GOSUB 35                                       <001>
2280 FOR I=1 TO 4:C$(I)="":NEXT:GOTO 50000         <026>
2499 REM**** DISK COMMAND ***********              <251>
2500 GOSUB 260:OPEN 1,8,15                         <242>
2510 PRINT"[CLEAR]":SYS P,12,23
   :PRINT"F1[SPACE]=[SPACE]MENUE"                  <212>
2520 PRINT"[HOME,SPACE8]-[SPACE]DISK[SPACE]
   COMMAND[SPACE]SENDEN -":C=0:Y=0                 <134>
2530 PRINT"[DOWN]>[SPACE]";:Y=Y+1                  <203>
2540 CX=2:CY=PEEK(214):BC=32:G=32:H=127:L=76
   :GOSUB 50:ON A GOTO 2560,2550                   <013>
2550 GOTO 50000                                    <125>
2560 PRINT#1,B$;                                   <160>
2570 INPUT#1,A,B$,B,C                              <165>
2580 PRINT:PRINT"[DOWN]"A;B$;B;C                   <188>
2590 IF Y<5 THEN 2530                              <101>
2600 GOSUB 35:GOTO 2510                            <178>
2999 REM**** DIRECTORY VON DISK *******            <242>
3000 GOSUB 260:OPEN 2,8,15,"I":OPEN 1,8,0,"$"
                              <000>
3010 C=-1:O$=CHR$(0):S$=CHR$(34)                   <039>
3020 PRINT"[CLEAR]":SYS P,12,22
   :PRINT"F1[SPACE]=[SPACE]MENUE"                  <211>
3030 X=0:PRINT"[HOME,SPACE]NR.[SPACE2]-[SPACE]
   ALTE[SPACE]REIHENFOLGE[SPACE]-[DOWN]"           <204>
3040 IF C=-1 THEN GET#1,A$,B$:X=-1                 <038>
3050 GET#1,A$,B$                                   <013>
3060 GET#1,A$,B$:X=X+1:C=C+1                       <219>
3070 B=ASC(A$+O$)+ASC(B$+O$)*256                   <035>
3080 PRINT C":"TAB(5);STR$(B);TAB(10);             <164>
3090 GET#1,B$:IF ST THEN 3190                      <133>
3100 IF B$<>S$THEN 3090                            <103>
3110 GET#1,B$:IF B$<>S$THEN PRINT B$;:GOTO 3110
                              <040>
3120 GET#1,B$:IF B$="[SPACE]"THEN 3120             <081>
3130 C$=""                                         <163>
3140 C$=C$+B$:GET#1,B$:IF B$<>""THEN 3140          <227>
3150 PRINT TAB(27);"[SPACE2]"LEFT$(C$,5)           <136>
3160 GET A$:IF A$="[F1]"THEN 3230:REM F1           <190>
3170 IF X=16 THEN 3200                             <214>
3180 IF ST=0 THEN 3050                             <251>
3190 PRINT"[RVSON,SPACE]BLOCKS[SPACE]FREE[SPACE,
   RVOFF]":GOSUB 35:GOTO 3230                      <157>
3200 SYS P,4,20:PRINT"BITTE[SPACE]TASTE[SPACE]
   DRUECKEN[SPACE]..."                             <083>
3210 SYS P,9,22:PRINT"F1[SPACE]=[SPACE]MENUE"
                              <190>
3220 GOSUB 35:IF A<>133 THEN 3030                  <003>
3230 GOTO 50000                                    <040>
3999 REM****** SCHREIBSCHUTZ *********             <190>
4000 PRINT"[CLEAR,SPACE11]-[SPACE]SCHREIBSCHUTZ
   [SPACE]-[DOWN]"                                 <107>
4010 GOSUB 200:PRINT"[DOWN,SPACE8,RVSON]"DN$"
   [RVOFF]"                                        <158>
4020 PRINT#1,"M-R";CHR$(1);CHR$(1);CHR$(1)         <084>
4030 GET#1,A$:IF A$="A"OR A$=""THEN 4130           <046>
4040 PRINT"[DOWN4]DISKETTE[SPACE]GESCHUETZT
   [SPACE]!"                                       <124>
4050 PRINT"[DOWN2]SOLL[SPACE]DIE[SPACE]DISKETTE
   [SPACE]FREIGEGEBEN[SPACE]WERDEN[SPACE]?"
                              <096>
4060 CY=18:GOSUB 20000                             <091>
4070 IF F=2 THEN 50000                             <070>
4080 PRINT#1,"M-W";CHR$(1);CHR$(1);CHR$(1);
   CHR$(65)                                        <083>
4090 PRINT#1,"B-P[SPACE]2[SPACE]2"
   :PRINT#2,CHR$(65)                               <024>
4100 PRINT#1,"B-P[SPACE]2[SPACE]165"
   :PRINT#2,CHR$(50);CHR$(65);                     <128>
4110 PRINT#1,"U2[SPACE]2[SPACE]0[SPACE]18[SPACE]
   0"                                              <217>
4120 GOTO 4200                                     <119>
4130 PRINT"[DOWN4]DISKETTE[SPACE]NICHT[SPACE]
   GESCHUETZT[SPACE]!"                             <077>
```

```
4140 PRINT"[DOWN2]SOLL[SPACE]DIE[SPACE]DISKETTE
     [SPACE]GESCHUETZT[SPACE]WERDEN[SPACE]?" <174>
4150 CY=18:GOSUB 20000                       <182>
4160 IF F=2 THEN 50000                       <161>
4170 PRINT#1,"B-P[SPACE]2[SPACE]2"
     :PRINT#2,CHR$(66)                       <106>
4180 PRINT#1,"B-P[SPACE]2[SPACE]165"
     :PRINT#2,CHR$(60);CHR$(60);             <204>
4190 PRINT#1,"U2[SPACE]2[SPACE]0[SPACE]18[SPACE]
     0"                                      <041>
4200 SYS P,14,23:PRINT"-[SPACE]OK[SPACE]-"  <147>
4210 GOSUB 35:GOTO 50000                     <047>
4499 REM**** DIRECTORY BEARBEITEN *****      <051>
4500 P$="[HOME,SPACE6]-[SPACE]DIRECTORY[SPACE]
     BEARBEITEN[SPACE]-[DOWN]"               <019>
4510 GOSUB 100:GOTO 4530:REM SEITE 1 NEU     <184>
4520 GOSUB 110:REM SEITE S NEU               <086>
4530 GOSUB 35                                <184>
4540 IF A=133 THEN 50000                     <126>
4550 IF A=17 THEN S=S+1+(S*16>Q):GOTO 4520  <136>
4560 IF A=145 THEN S=S-1-(S=0):GOTO 4520    <147>
4570 IF A=134 THEN 5000:REM SORTIEREN        <241>
4580 IF A=135 THEN 6000:REM EINFUEGEN        <216>
4590 IF A=136 THEN 7000:REM LOESCHEN         <159>
4600 GOTO 4530                               <094>
4999 REM********** SORTIEREN **********      <044>
5000 GOSUB 21000                             <027>
5010 CX=34:CY=7:L=3:G=48:H=57:BC=46:GOSUB 50
     :ON A GOTO 5020,4520                    <028>
5020 F=VAL(B$)                               <035>
5030 IF F>Q THEN F=Q                         <124>
5040 IF F<1 THEN F=1                         <072>
5050 CX=34:CY=9:GOSUB 50:ON A GOTO 5060,4520
                          <197>
5060 R=VAL(B$)                               <087>
5070 IF R>Q THEN R=Q                         <188>
5080 IF R<1 THEN R=1                         <136>
5090 GOSUB 5110:GOTO 4520                    <215>
5100 REM --------------------                <229>
5110 A$=N$(F)                                <041>
5120 IF F>R THEN 5160                        <091>
5130 FOR I=F TO R-1:REM F<R                  <079>
5140 N$(I)=N$(I+1)                           <205>
5150 NEXT I:N$(R)=A$:RETURN                  <043>
5160 FOR I=F TO R+1 STEP-1:REM F>R           <243>
5170 N$(I)=N$(I-1)                           <236>
5180 NEXT I:N$(R)=A$:RETURN                  <073>
5999 REM********* EINFUEGEN ***********      <243>
6000 A$="EINFUEGEN":GOSUB 21500              <055>
6010 CX=33:CY=7:L=3:G=48:H=57:BC=46:GOSUB 50
     :ON A GOTO 6020,4520                    <008>
6020 R=VAL(B$)                               <027>
6030 IF R>144 THEN 6010                      <058>
6040 IF R<1 THEN R=1                         <076>
6050 SYS P,6,20:PRINT"TEXT[SPACE]DER[SPACE]
     BEMERKUNG[SPACE]?"                      <173>
6060 SYS P,7,21:PRINT"----------------"      <138>
6070 CX=7:CY=21:L=16:G=5:H=255:BC=45:GOSUB 50
     :ON A GOTO 6080,4520                    <115>
6080 IF B$=""THEN B$="----------------"      <145>
6090 IF LEN(B$)<16 THEN FOR I=1 TO 16-LEN(B$)
     :B$=B$+"[SPACE]":NEXT                   <230>
6100 B$=CHR$(130)+LEFT$(FR$,2)+B$+RIGHT$(FR$,11)
                          <170>
6110 IF R>Q THEN 6150                        <071>
6120 IF N$(R)=FR$THEN N$(R)=B$:GOTO 4520    <111>
6130 IF N$(Q)=FR$THEN F=Q:N$(F)=B$:GOSUB 5110
     :GOTO 4520                              <125>
6140 IF Q<145 THEN F=Q+1:N$(F)=B$:GOSUB 5110
     :GOTO 6170                              <129>
6150 IF Q>136 THEN 6010                      <179>
6160 N$(R)=B$                                <085>
6170 A=Q+1                                   <081>
6180 Q=Q+8:IF R>Q THEN 6180                  <001>
6190 FOR I=A TO Q:IF N$(I)=""THEN N$(I)=FR$
                          <224>
6200 NEXT I                                  <027>
6210 GOTO 4520                               <174>
6999 REM********** LOESCHEN ***********      <196>
7000 A$="LOESCHEN":GOSUB 21500               <222>
7010 CX=33:CY=7:L=3:G=48:H=57:BC=46:GOSUB 50
     :ON A GOTO 7020,4520                    <245>
7020 F=VAL(B$):R=Q                           <138>
7030 IF F>Q OR F<1 THEN 7000                 <172>
7040 GOSUB 5110                              <239>
7050 N$(Q)=FR$                               <039>
7060 A=Q-7:G=0                               <056>
7070 FOR I=A TO Q:IF N$(I)<>FR$THEN G=1      <045>
7080 NEXT I                                  <142>
7090 IF G=0 THEN Q=Q-8:N$(Q+1)="":GOTO 7060
                          <014>
7100 GOTO 4520                               <043>
17999 REM**** ENTSCHEIDUNG VERT ******       <138>
18000 PRINT P$                               <163>
18010 F=1:SYS P,12,24:PRINT"F1[SPACE]=[SPACE]
      MENUE";                                <146>
18020 SYS P,CX,5:C$(F)="[RVSON]"             <000>
18030 PRINT C$(1)C$(3)                       <033>
18040 SYS P,CY,7                             <215>
18050 PRINT C$(2)C$(4)                       <055>
18060 C$(1)="":C$(2)=""                      <203>
18070 GOSUB 35                               <209>
18080 IF A=17 THEN F=2:GOTO 18020            <091>
18090 IF A=145 THEN F=1:GOTO 18020           <150>
18100 IF A=133 THEN C$(3)="":C$(4)="":GOTO 50000
                          <107>
18110 IF A<>13 THEN 18070                    <063>
18120 C$(3)="":C$(4)=""                      <011>
18130 RETURN                                 <166>
19999 REM**** ENTSCHEIDUNG HORIZ ******      <173>
20000 F=1                                    <119>
20010 C$(F)="[RVSON]"                        <126>
20020 SYS P,10,CY:PRINT C$(1)"[SPACE]JA[SPACE,
      RVOFF]":SYS P,20,CY:PRINT C$(2)"[SPACE]NEIN
      [SPACE,RVOFF]"                         <029>
20030 C$(1)="":C$(2)=""                      <133>
20040 GOSUB 35                               <139>
20050 IF A=29 OR A=78 THEN F=2:GOTO 20010    <034>
20060 IF A=157 OR A=74 THEN F=1:GOTO 20010   <089>
20070 IF A=13 THEN RETURN                    <203>
20080 IF A=133 THEN C$(1)="":C$(2)="":GOTO 50000
                          <043>
20090 GOTO 20040                             <071>
20999 REM****** MASKE SORTIEREN *******      <054>
21000 PRINT"[HOME,DOWN3]";TAB(28);"BITTE[SPACE]
      NR."                                   <179>
21010 PRINT TAB(28);"EINGEBEN"               <239>
21020 PRINT TAB(28);"[SPACE9]"               <188>
21030 PRINT TAB(28);"FILE"                   <230>
21040 PRINT TAB(28);"NR.:[SPACE2]..."        <098>
21050 PRINT TAB(28);"[SPACE9]"               <218>
21060 PRINT TAB(28);"NACH:[SPACE]..."        <194>
21070 PRINT TAB(28);"[SPACE9]"               <238>
21080 PRINT TAB(28);"SORTIEREN"              <179>
21090 FOR I=1 TO 8:PRINT TAB(28);"[SPACE11]"
      :NEXT:RETURN                           <073>
21499 REM**** MASKE EINF/LOESCH *******      <043>
21500 PRINT"[HOME,DOWN3]"TAB(28);"BEMERKUNG"
                          <199>
21510 PRINT TAB(28);A$                       <201>
21520 PRINT TAB(28);"AN[SPACE7]"             <065>
21530 PRINT TAB(28);"POSITION[SPACE]"        <049>
21540 PRINT TAB(28);"NR.:[SPACE]...[SPACE]"
                          <088>
21560 FOR I=1 TO 12:PRINT TAB(28);"[SPACE11]"
      :NEXT:RETURN                           <076>
21999 REM**** MASKE LETZTE ZEILE ******      <057>
22000 PRINT"[CLEAR]"                         <181>
22010 SYS P,0,22                             <041>
22020 PRINT"[SPACE]@@@@@@@@@@@"              <159>
22030 PRINT"[SPACE,RVSON]BLAETTERN[SPACE,RVOFF]"
                          <134>
22040 PRINT"[SPACE,RVSON]MIT[SPACE]CURSOR[RVOFF]
      ";                                     <242>
22050 SYS P,14,24:PRINT"F1[SPACE]=[SPACE]MENUE";
                          <009>
22060 SYS P,26,21:PRINT"[SPACE]@@@@@@@@@@@" <018>
22080 SYS P,26,22:PRINT"[SPACE,RVSON]F3=SORTIER
      [RVOFF]"                               <065>
22090 SYS P,26,23:PRINT"[SPACE,RVSON]F5=EINFUEG
      [RVOFF]"                               <041>
22100 SYS P,26,24:PRINT"[SPACE,RVSON]F7=LOESCH
      [SPACE,RVOFF]";                        <044>
22110 PRINT"[HOME]"P$:RETURN                 <224>
29999 REM*********** ENDE **************     <063>
30000 PRINT"[CLEAR]"                         <021>
30010 SYS P,4,12:PRINT"WOLLEN[SPACE]SIE[SPACE]
      WIRKLICH[SPACE]BEENDEN[SPACE]?"        <226>
30020 CY=15:GOSUB 20000                      <038>
30030 ON F GOTO 30040,50000                  <059>
30040 PRINT"[CLEAR]":END                     <247>
49999 REM******* HAUPTMENUE ***********      <145>
```

```
50000 CLOSE 2:CLOSE 1                                    <240>
50005 PRINT"[CLEAR,SPACE8]****[SPACE4]
   U**************I"                                     <196>
50010 PRINT "[SPACE8]B[SPACE]O[SPACE]B[SPACE4]B
   [SPACE2]TRACK[SPACE]18[SPACE2]B"                      <047>
50020 PRINT "[SPACE8]B.W[SPACE]B[SPACE4]
   J**************K"                                     <034>
50030 PRINT "[SPACE8]BT[SPACE]↑B"                        <146>
50040 PRINT "[SPACE8]Z***X"                              <194>
50050 SYS P,13,6:C$(U)="[RVSON]"
   :PRINT"-[SPACE]BY[SPACE]MATAN[SPACE]-"                <241>
50060 PRINT"[SPACE6]******************************"
   :PRINT                                                <063>
50070 PRINT"[SPACE7]"C$(1)"DIRECTORY[SPACE]
   EINLESEN[RVOFF]"                                      <253>
50080 PRINT"[SPACE7]"C$(2)"DIRECTORY[SPACE]
   BEARBEITEN[RVOFF]"                                    <134>
50090 PRINT"[SPACE7]"C$(3)"NEUES[SPACE]
   DIRECTORY[SPACE]SCHREIBEN[RVOFF]"                     <211>
50100 PRINT"[SPACE7]"C$(4)"DIRECTORY[SPACE]VON
   [SPACE]DISK[RVOFF]"                                   <233>
50110 PRINT                                              <026>
50120 PRINT"[SPACE7]"C$(5)"DISK[SPACE]COMMAND
   [SPACE]SENDEN[RVOFF]"                                 <018>
50130 PRINT"[SPACE7]"C$(6)"DISKETTENNAMEN[SPACE]
   AENDERN[RVOFF]"                                       <077>
50140 PRINT"[SPACE7]"C$(7)"DISK-KOMMENTAR[RVOFF]
   "                                                     <071>
50150 PRINT"[SPACE7]"C$(8)"FILES[SPACE]
   SCHUETZEN/FREIGEBEN[RVOFF]"                           <046>
50160 PRINT"[SPACE7]"C$(9)"SCHREIBSCHUTZ[RVOFF]"
                          <056>
50200 PRINT                                              <117>
50210 PRINT"[SPACE6]******************************"
                          <003>
50220 PRINT"[SPACE7]"C$(10)"[SPACE]*[SPACE]ENDE
   [SPACE]*[SPACE,RVOFF]"                                <044>
50230 SYS P,11,3:POKE 207,0:POKE 204,0                   <094>
50240 GOSUB 35:POKE 204,1:C$(U)=""                       <250>
50250 IF A=17 THEN U=1-(U<10)*U                          <057>
50260 IF A=145 THEN U=U-1-10*(U=1)                       <080>
50270 IF A<>13 THEN 50050                                <087>
50280 ON U GOTO 300,4500,1000,3000,2500,1500,
   2000,600,4000,30000                                   <163>
50290 GOTO 50230                                         <185>
```

Disketten-Meister

Um leichter mit der Diskette arbeiten zu können, benötigen Sie nur diese kleine Basic-Erweiterung.

Die hier vorgestellte Basic-Erweiterung umfaßt 24 neue Befehle, die es ermöglichen, den Umgang mit dem Diskettenlaufwerk sowohl im Direktmodus als auch im Programm erheblich zu vereinfachen. Autostart, menügesteuertes Laden von Programmen sowie das Speichern von Maschinenprogrammen sind nur 3 der vielen Möglichkeiten dieses Programms.

Die Befehle im Einzelnen:

@
Der Fehlerkanal des Laufwerks wird gelesen und auf dem Bildschirm angezeigt.

@S
Der Fehlerkanal wird gelesen und die Nummer des Fehlers in Speicherstelle 251 abgelegt. Die Abfrage erfolgt nun mit PRINT PEEK (251). Die Bedeutung der Fehlernummern ist aus dem Floppy-Bedienungshandbuch Seite 36 bis 39 zu entnehmen.

$
Das Inhaltsverzeichnis der Diskette wird auf dem Bildschirm seitenweise angezeigt, das heißt daß die Ausgabe nach 23 Zeilen stoppt und das Programm eine Menüzeile ausgibt. Nun hat man die Möglichkeit die folgenden Zeilen zu listen oder die Ausgabe zu beenden.

$$
Dieser Befehl erlaubt es auf einfachste Art ein Programm zu laden ohne den Namen des Programms einzugeben. Hier wird das Inhaltsverzeichnis der Diskette gelistet, wobei jeder Zeile eine Programmnummer vorangestellt ist. Sind 20 Zeilen ausgegeben, so erscheint eine Menüzeile. Man kann nun die nächste Seite listen, die Ausgabe beenden, oder ein Programm laden. Drückt man nun die Taste für »Laden«, so ist nur die Nummer des Programms einzugeben, »RETURN« zu drücken und das jeweilige Programm wird in den Basic-Speicher geladen. Es besteht die Möglichkeit durch ein der Programmnummer nachgestelltes &-Zeichen einen Autostart durchzuführen.

#
Die aktuelle Geräteadresse wird auf dem Bildschirm angezeigt. Sie wird beim Start des Programms automatisch auf 8 gesetzt.

#9
Ändern der Geräteadresse. Alle folgenden Befehle beziehen sich auf Geräteadresse 9.

#8
Alle Befehle beziehen sich wieder auf Geräteadresse 8.

#E
Die Befehlserweiterung wird ausgeschaltet. Dies ist notwendig, wenn man ein anderes Maschinenprogramm in den Speicher ab $C000 laden will, ohne daß das System aussteigt. Alle Zeiger werden wieder in ihre Ausgangsposition gesetzt.

£"Name"
Die Startadresse des jeweiligen Programms wird auf dem Bildschirm angezeigt. Dieser Befehl kann bei unbekannten Startadressen von Maschinenprogrammen eine nützliche Hilfe sein.

%"Name"
Das Programm mit dem angegebenen Namen wird geladen. Dieser Befehl lädt ein Programm immer an die Adresse, die im Programm abgespeichert ist. Es lassen sich so auch Maschinenprogramme laden.

&"Name"
Autostart-Befehl. Das Programm wird geladen und danach selbständig gestartet. Dies ist nur mit Programmen möglich, die im Basic-Speicher abgelegt sind.

!"Name"
Das im Speicher befindliche Basic-Programm wird unter dem angegeben Namen auf Diskette gespeichert.

!"Name",S,E
Dieser Befehl ermöglicht das Speichern von Maschinenprogrammen, wobei »S« die Startadresse und »E« die Endadresse ist. Es muß beachtet werden, daß »E« die Endadresse + 1 des abzuspeichernden Bereichs ist.

←V
Der Validate-Befehl wird ausgeführt.

←I
Initialisiert eine Diskette.

←N"Name,ID"
Formatiert eine Diskette.

←N"Name"
Reformatiert eine Diskette ohne Änderung der ID. Dies funktioniert jedoch nur bei bereits formatierten Disketten.

←C"nf=af"
Das File »af« wird mit dem Namen »nf« kopiert.

←C"nf=file1,file2"
Es besteht die Möglichkeit maximal 4 sequentielle Datenfiles zu einem mit neuem Namen zusammenzufassen.

←R"nfn=afn"
Das File mit dem Namen »afn« wird zu »nfn« umbenannt.

←S"file"
Das File mit dem Namen »file« wird gelöscht.

←S"file1,file2,..."
Es können auch mehrere Files gleichzeitig gelöscht werden.

←S"fi*"
Es werden alle Files gelöscht, die mit dem Buchstaben »fi« beginnen.

←S"*"
Alle Files einer Diskette werden gelöscht.

Eingabe des Programms
Das Basic-Programm bildet jeweils von fünf DATA-Zeilen eine Prüfsumme. Ist diese fehlerhaft, so stoppt das Programm und zeigt die jeweilige Blocknummer auf dem Bildschirm an. Waren die Prüfsummen aller 20 Blocks fehlerfrei, stoppt das Programm mit dem Hinweis die richtige Diskette einzulegen. Nach Drücken einer Taste wird das Maschinenprogramm unter dem Namen »Disk-Master/M« auf Diskette gespeichert. Um das Laden und Starten zu erleichtern, muß der Basic-Lader eingetippt werden.

Laden des Programms
mit Basic-Lader:
LOAD"Disk-Master",8. Ist die Ready-Meldung erfolgt, so ist einfach RUN einzugeben und das Maschinenprogramm wird geladen und gestartet.
direkt:
LOAD"Disk-Master/M",8,1. Ist die Ready-Meldung erfolgt, muß das Programm mit SYS 49152 gestartet werden.

Allgemeine Hinweise zur Programmbenutzung:
Will man im Rahmen eines Basic-Programmes einen Befehl hinter einer »IF...THEN« Abfrage benutzen, so ist vor dem Befehl ein Doppelpunkt zu setzen, da sonst eine Fehlermeldung ausgegeben wird.
Sollen die Befehle ←N, ←R, ←S, ←C im Programm verwendet werden, so sollten die nachfolgenden Parameter in einer String-Variablen gespeichert sein.
Will man im Anschluß an »Disk-Master« noch andere Programme abspeichern, so darf die Startadresse des folgenden Programms nicht kleiner als $C900 sein.

(Armin Haas/rg)

```
100 REM ********  BASIC-LADER  ********     <138>
110 IF F=1 THEN SYS 49152:NEW               <072>
120 PRINT"[DOWN]LOADING[SPACE]DISK-MASTER/M"
                                     <004>
130 F=1:LOAD"DISK-MASTER/M",8,1             <029>
```

Befehlsübersicht »Disk-Master«

Befehl	Funktion
@	Fehlerkanal lesen und anzeigen
@S	Fehlerkanal lesen und speichern
$	Inhaltsverzeichnis der Diskette anzeigen
$$	Menügesteuertes Laden von Programmen
#	Aktuelle Geräteadresse abfragen
#9	Geräteadresse = 9
#8	Geräteadresse = 8
#E	Programm ausschalten
£"Name"	Startadresse ermitteln und anzeigen
%"Name"	Programm absolut laden
&"Name"	Programm laden und starten
!"Name"	Programm speichern
!"Name",A,B	Maschinenprogramm speichern
←V	Validate-Befehl ausführen
←I	Initialisiert Diskette
←N"Name,ID"	Formatiert Diskette
←N"Name"	Reformatiert Diskette
←C"nf=af"	Kopiert File
←C"nf=file1,file2"	Verkettet SEQ-Files (max. 4)
←R"nfn=afn"	File umbenennen
←S"file"	File löschen
←S"file1,file2,..."	Mehrere Files gleichzeitig löschen
←S"fi*"	Alle Files mit Anfangsbuchstaben »fi« werden gelöscht
←S"*"	Alle Files der Diskette werden gelöscht

```
0 REM *************************                <169>
1 REM *                       *                <228>
2 REM *    --DISK-MASTER--    *                <189>
4 REM *                       *                <231>
5 REM *      1984 ARMIN HAAS  *  **            <166>
6 REM *                       *                <233>
7 REM *************************                <176>
8 REM                                          <151>
10 REM PRUEFSUMMENTEST                         <048>
20 N=1                                         <069>
30 S=0                                         <083>
40 FOR I=1 TO 80                               <225>
50 READ X:IF X=-1 THEN 80                      <203>
60 S=S+X                                       <150>
70 NEXT                                        <200>
80 READ F:IF F<>S THEN 120                     <025>
90 PRINT"PRUEFSUMME[SPACE]BLOCK"N"[SPACE]OK!"
                       <248>
100 IF X=-1 THEN 140                           <017>
110 N=N+1:GOTO 30                              <189>
120 PRINT"FEHLER[SPACE]IN[SPACE]BLOCK"N:END
                       <021>
130 REM MASCHINENPROGRAMM WIRD AUF DISKETTE GES
   PEICHERT                                    <174>
140 PRINT"[DOWN]DISKETTE[SPACE]EINLEGEN[SPACE]
   UND[SPACE]TASTE[SPACE]DRUEKEN!"             <181>
150 WAIT 198,1                                 <039>
160 PRINT"[DOWN]PROGRAMM[SPACE]WIRD[SPACE]AUF
   [SPACE]DISK[SPACE]GESPEICHERT"
   :PRINT"[DOWN]BITTE[SPACE]WARTEN!"           <245>
170 OPEN 1,8,1,"DISK-MASTER/M"                 <075>
180 PRINT#1,CHR$(0);:REM ADRESSE LOW           <238>
190 PRINT#1,CHR$(192);:REM ADRESSE HIGH        <146>
200 RESTORE                                    <084>
210 FOR I=1 TO 1585                            <246>
220 READ X:IF X>255 THEN 240                   <098>
230 PRINT#1,CHR$(X);                           <134>
240 NEXT I                                     <187>
250 CLOSE 1                                    <203>
260 PRINT"[DOWN]FERTIG[SPACE]!":END            <143>
495 REM DATAS                                  <236>
500 DATA 169,0,141,32,208,141,33,208,169,15,141,
   134,2,169,74,162                            <017>
510 DATA 56,160,192,32,26,192,32,45,192,96,133,
   253,134,251,132,252                         <134>
520 DATA 160,0,177,251,32,210,255,200,196,253,
   208,246,96,169,101,141                      <236>
530 DATA 8,3,169,194,141,9,3,96,147,17,42,42,42,
   42,42,42                                    <173>
540 DATA 42,42,42,42,42,32,32,68,73,83,75,32,45,
   45,32,77                                    <182>
545 DATA 8514:REM PRUEFSUMME BLOCK 1           <230>
550 DATA 65,83,84,69,82,32,32,42,42,42,42,42,42,
   42,42,42                                    <186>
560 DATA 42,42,17,29,29,29,29,29,29,29,29,29,29,
   40,67,41                                    <225>
570 DATA 32,49,57,56,52,32,32,65,82,77,73,78,32,
   72,65,65                                    <237>
580 DATA 83,17,17,18,32,32,60,70,49,62,32,87,69,
   73,84,69                                    <245>
590 DATA 82,32,32,32,60,70,51,62,32,69,78,68,69,
   32,32,146                                   <032>
595 DATA 4084:REM PRUEFSUMME BLOCK 2           <023>


600 DATA 8,169,1,174,160,192,160,15,32,186,255,
   96,169,0,32,189                             <044>
610 DATA 255,32,192,255,162,1,32,198,255,96,32,
   207,255,32,210,255                          <190>
620 DATA 165,144,240,246,32,204,255,169,1,32,
   195,255,96,169,36,133                       <003>
630 DATA 251,169,251,133,187,169,0,133,188,169,
   1,133,183,173,160,192                       <112>
640 DATA 133,186,169,96,133,185,32,213,243,165,
   186,32,180,255,165,185                      <182>
645 DATA 11729:REM PRUEFSUMME BLOCK 3          <126>
650 DATA 32,150,255,169,0,133,144,96,160,3,132,
   251,32,165,255,133                          <213>
660 DATA 252,164,144,208,56,32,165,255,164,144,
   208,49,164,251,136,208                      <190>
670 DATA 233,166,252,32,205,189,169,32,32,210,
   255,32,165,255,166,144                      <143>
680 DATA 208,27,170,240,6,32,210,255,76,27,193,
   169,13,32,210,255                           <199>
690 DATA 164,254,192,22,240,11,200,132,254,160,
   2,208,189,32,66,246                         <046>
695 DATA 11742:REM PRUEFSUMME BLOCK 4          <172>
700 DATA 96,169,0,133,254,169,30,162,130,160,
   192,32,26,192,32,228                        <019>
710 DATA 255,240,251,201,133,240,7,201,134,240,
   15,76,78,193,169,147                        <124>
720 DATA 32,210,255,169,0,133,254,76,57,193,169,
   13,32,210,255,169                           <045>
730 DATA 23,32,210,255,76,61,193,32,161,192,169,
   1,162,255,160,159                           <050>
740 DATA 32,189,255,76,165,193,32,161,192,32,87,
   226,165,187,56,233                          <140>
745 DATA 11234:REM PRUEFSUMME BLOCK 5          <219>
750 DATA 2,133,187,160,0,165,253,145,187,200,
   169,58,145,187,165,183                      <184>
760 DATA 24,105,2,133,183,32,192,255,169,1,32,
   195,255,96,165,251                          <035>
770 DATA 56,233,48,133,251,169,0,160,10,24,101,
   251,136,208,250,133                         <116>
780 DATA 251,165,252,56,233,48,24,101,251,133,
   251,96,32,161,192,32                        <142>
790 DATA 172,192,76,186,192,169,147,32,210,255,
   169,0,133,254,32,205                        <220>
795 DATA 11294:REM PRUEFSUMME BLOCK 6          <021>
800 DATA 192,76,248,192,32,161,192,32,172,192,
   32,207,255,133,251,32                       <191>
810 DATA 207,255,133,252,32,207,255,165,144,240,
   249,32,196,192,76,174                       <087>
820 DATA 193,169,228,141,8,3,169,167,141,9,3,96,
   169,0,174,160                               <218>
830 DATA 192,32,205,189,169,13,32,210,255,96,
   169,2,174,160,192,160                       <209>
840 DATA 2,32,186,255,96,165,183,166,187,164,
   188,32,189,255,32,192                       <248>
845 DATA 11612:REM PRUEFSUMME BLOCK 7          <066>
850 DATA 255,162,2,32,198,255,32,207,255,133,
   251,32,207,255,133,252                      <009>
860 DATA 32,204,255,169,2,32,195,255,96,32,26,
   194,32,87,226,169                           <103>
870 DATA 0,133,251,133,252,32,46,194,165,252,
   166,251,32,205,189,169                      <036>
880 DATA 13,32,210,255,96,32,115,0,201,36,240,
   73,201,64,240,94                            <025>
890 DATA 201,95,240,53,201,35,240,105,201,38,
   240,36,201,37,240,23                        <175>
895 DATA 11225:REM PRUEFSUMME BLOCK 8          <117>
900 DATA 201,92,240,10,201,33,240,36,32,121,0,
   76,231,167,32,115                           <081>
910 DATA 0,32,73,194,76,174,167,32,115,0,32,69,
   195,32,174,167                              <042>
920 DATA 32,115,0,32,133,195,76,174,167,76,16,
   195,32,115,0,32                             <036>
930 DATA 156,195,76,174,167,32,115,0,201,36,240,
   6,32,213,193,76                             <151>
940 DATA 174,167,32,115,0,76,212,195,32,115,0,
   76,174,167,32,115                           <156>
945 DATA 8343:REM PRUEFSUMME BLOCK 9           <127>
950 DATA 0,201,83,240,6,32,204,193,76,174,167,
   32,228,193,76,200                           <166>
960 DATA 194,32,115,0,240,24,201,69,240,32,32,
   158,183,224,8,240                           <162>
970 DATA 19,224,9,240,15,224,1,240,11,162,11,32,
   55,164,32,12                                <009>
980 DATA 194,76,174,167,142,160,192,76,174,167,
   32,1,194,76,200,194                         <117>
990 DATA 32,115,0,201,86,240,26,201,73,240,22,
   201,82,240,27,201                           <169>
995 DATA 9754:REM PRUEFSUMME BLOCK 10          <224>
1000 DATA 83,240,23,201,67,240,19,201,78,240,15,
   32,121,0,76,231                             <146>
1010 DATA 167,141,255,159,32,119,193,76,200,194,
   133,253,32,115,0,32                         <119>
H    195,96,174,160,192,160                    <210
1030 DATA 1,32,186,255,32,87,226,169,0,32,213,
   255,144,3,76,249                            <159>
1040 DATA 224,134,45,132,46,32,183,255,41,191,
   240,5,162,29,76,55                          <007>
1045 DATA 9948:REM PRUEFSUMME BLOCK 11         <025>
1050 DATA 164,96,164,56,133,51,132,52,165,45,
   164,46,133,47,132,48                        <077>
1060 DATA 133,49,132,50,96,174,160,192,160,0,32,
   186,255,32,87,226                           <081>
1070 DATA 169,0,166,43,164,44,32,89,195,76,113,
   168,174,160,192,32                          <106>
1080 DATA 186,255,32,87,226,32,121,0,240,39,32,
   253,174,32,138,173                          <094>

1090 DATA 32,247,183,165,20,133,251,165,21,133,
   252,32,253,174,32,138                       <243>
```

```
1095 DATA 9660:REM PRUEFSUMME BLOCK 12     <067>
1100 DATA 173,32,247,183,169,251,166,20,164,21,
   134,253,132,254,76,95                   <019>
1110 DATA 225,76,89,225,169,0,133,253,169,199,
   133,254,169,0,141,250                   <244>
1120 DATA 198,141,251,198,141,252,198,141,253,
   198,169,147,32,210,255,32               <197>
1130 DATA 205,192,160,3,132,251,32,165,255,133,
   252,164,144,208,117,32                  <073>
1140 DATA 165,255,164,144,208,110,164,251,136,
   208,233,166,252,32,205,189              <250>
1145 DATA 12998:REM PRUEFSUMME BLOCK 13    <174>
1150 DATA 169,32,32,210,255,32,165,255,166,144,
   208,88,170,240,9,32                     <219>
1160 DATA 210,255,32,123,196,76,21,196,169,13,
   32,210,255,152,160,0                    <212>
1170 DATA 145,253,76,158,196,169,18,32,210,255,
   32,180,197,169,0,174                    <052>
1180 DATA 250,198,232,142,250,198,32,205,189,
   169,146,32,210,255,169,32               <201>
1190 DATA 32,210,255,169,45,32,210,255,169,32,
   32,210,255,165,253,24                   <039>
1195 DATA 11628:REM PRUEFSUMME BLOCK 14    <214>
1200 DATA 105,17,144,5,164,254,200,132,254,133,
   253,169,0,141,252,198                   <098>
1210 DATA 160,2,208,128,32,11,197,32,66,246,96,
   201,34,208,16,174                       <174>
1220 DATA 253,198,240,5,162,0,76,139,196,162,1,
   142,253,198,96,174                      <001>
1230 DATA 253,198,240,250,172,252,198,200,140,
   252,198,145,253,96,173,250              <094>
1240 DATA 198,201,20,240,3,76,53,196,169,37,162,
   203,160,197,32,197                      <062>
1245 DATA 11941:REM PRUEFSUMME BLOCK 15    <007>
1250 DATA 196,32,228,255,240,251,201,133,240,30,
   201,134,240,71,201,135                  <224>
1260 DATA 240,85,76,177,196,133,60,134,61,132,
   62,160,0,177,61,32                      <226>
1270 DATA 210,255,200,196,60,208,246,96,169,147,
   32,210,255,169,0,141                    <179>
1280 DATA 250,198,141,251,198,141,252,198,133,
   253,169,199,133,254,160,0               <099>
1290 DATA 169,32,153,0,199,200,208,250,160,0,
   153,0,200,200,192,90                    <019>
1295 DATA 12304:REM PRUEFSUMME BLOCK 16    <053>
1300 DATA 208,248,76,53,196,32,119,196,76,174,
   167,169,23,162,240,160                  <239>
1310 DATA 197,32,197,196,76,168,196,169,22,162,
   7,160,198,32,197,196                    <224>
1320 DATA 32,119,196,169,0,133,251,133,252,133,
   253,32,207,255,56,233                   <225>
1330 DATA 48,133,251,32,207,255,201,38,208,4,
   133,253,169,13,201,13                   <123>
1340 DATA 208,7,169,0,133,252,76,98,197,56,233,
   48,133,252,32,207                       <069>
1345 DATA 11218:REM PRUEFSUMME BLOCK 17    <107>
H    136,208,250,24,101,252                 <227>
1360 DATA 133,251,169,0,133,252,164,251,24,105,
   17,144,5,166,252,232                    <205>
1370 DATA 134,252,133,251,136,208,241,165,252,
   24,105,199,133,252,177,251              <220>
1380 DATA 166,251,232,208,5,164,252,200,132,252,
   164,252,32,189,255,165                  <131>
1390 DATA 253,201,38,240,17,174,160,192,160,1,
   32,186,255,32,87,195                    <202>
1395 DATA 12680:REM PRUEFSUMME BLOCK 18    <162>
1400 DATA 32,114,195,76,174,167,174,160,192,160,
   0,32,186,255,32,144                     <009>
1410 DATA 195,76,174,167,173,250,198,201,9,208,
   5,169,1,141,251,198                     <244>
1420 DATA 173,251,198,208,5,169,48,32,210,255,
   96,17,18,60,70,49                       <102>
1430 DATA 62,32,87,69,73,84,69,82,32,60,70,51,
   62,32,69,78                             <079>
1440 DATA 68,69,32,60,70,53,62,32,76,65,68,69,
   78,146,17,13                            <139>
1445 DATA 8358:REM PRUEFSUMME BLOCK 19     <172>
1450 DATA 145,157,157,157,157,157,32,32,32,32,
   157,157,157,157,17,32                   <070>
1460 DATA 32,32,32,157,157,157,157,78,85,77,77,
   69,82,32,68,69                          <070>
1470 DATA 83,32,80,82,79,71,82,65,77,77,83,63,
   32,13,-1                                <212>
1475 DATA 4015:REM PRUEFSUMME BLOCK 20     <180>
```

Fileprotect 64

Sicher kennen auch Sie das Problem: »OPEN1,8,15,"S:xyz*"« und schon ist es passiert! Wie leicht löscht man unbeabsichtigt ein Programm oder eine Datei von der Diskette.

Das Programm Fileproject 64 sorgt dafür, daß Ihnen so etwas nicht mehr passieren kann. Mit ihm lassen sich alle Filetypen der Floppy 1541 vorm Scratchen schützen. Natürlich kann der Schutz auch wieder entfernt werden.

Die Sprungadressen sind so angelegt, daß Sie beim Abtippen die REM-Zeichen nicht mit eingeben müssen.

Nach dem Starten des Programms erscheint das Titelbild mit dem Menü (Bild 1), von dem aus Sie in drei Unterprogramme verzweigen, oder das Programm beenden können. Drücken Sie hier die »F8-Taste«, so erzeugen Sie einen System-Reset (SYS 64738). Haben Sie Fileprotect 64 noch nicht abgespeichert, müssen Sie es mit der »STOP-Taste« beenden, sonst war die Arbeit des Abtippens umsonst!.

Nehmen wir an, Sie wollen das Directory einer Diskette einlesen. Betätigen Sie dazu bitte die »F1-Taste«. Der Bildschirm wird gelöscht, es erscheint die Überschrift »DIRECTORY« und Sie werden aufgefordert eine Diskette einzulegen und eine Taste zu drücken. Sollten Sie versehentlich eine Taste drücken und es befindet sich noch keine Disk im Laufwerk, so wird die Fehlermeldung »DISK FEHLER« ausgegeben (Bild 2) und ein erneuter Tastendruck erwartet. Nach Auflisten der Einträge müssen Sie wieder eine Taste drücken (Bild 3), um zum Menü zurückzukehren. Hat das Directory mehr als 15 Einträge, so erscheint die Meldung »WEITERE EINTRAEGE BITTE TASTE DRUECKEN« und es wird eine neue Seite angelegt und angezeigt.

Möchten Sie ein Programm schützen, betätigen Sie die »F3-Taste«. Jetzt werden Sie erneut aufgefordert, eine Diskette einzulegen und eine Taste zu drücken. Haben Sie dies getan, prüft Fileprotect 64, ob eine Disk eingelegt ist und ob sie mit einem Schreibschutz versehen ist! Ist dies der Fall, wird die Fehlermeldung »SCHREIBSCHUTZ ENTFERNEN« ausgegeben (Bild 4). Haben Sie eine korrekte Disk eingelegt, durchsucht das Programm die Directory nach ungeschützten Files; wurde ein solcher Eintrag gefunden, so wird dessen Filetyp und Name ausgegeben. Danach erscheint »PROTECT (J/N)« (Bild 5) und Sie können entscheiden, ob Sie das File schützen wollen oder ob nicht. Sind keine weiteren ungeschützten Files auf der Diskette, kehrt das Programm zum Menü zurück. Drücken Sie jetzt die »F1-Taste« und lesen das Directory ein, so erkennen Sie geschützte Files an dem >-Zeichen hinter dem Filetyp. Wählen Sie erneut »Protect« an, werden Sie feststellen, daß die eben geschützten Files nicht mehr abgefragt werden! Möchten Sie den Scratchschutz wieder entfernen, wählen Sie »F5 UNPROTECT«. Alles weitere läuft dann so ab, wie Sie es von »Protect« kennen.

Nun möchte ich noch einige Worte über die Funktionsweise von »Fileprotect 64« sagen.

Das Betriebssystem der Floppy 1541 kennt, wie Sie sicher wissen, fünf verschiedene Filetypen. Diese können geöffnet, geschlossen oder geschützt sein. Somit sind dann 15 verschiedene Zustände möglich. Der Filetyp wird in der Directory durch das erste Byte eines Fileeintrages gekennzeichnet. Wie

Sie in Tabelle 1 sehen, unterscheidet sich ein geschützter Filetyp von einem ungeschützten durch das gesetzte sechste Bit. Das Setzen des Bits erreicht man, indem man den ungeschützten Filetyp durch die »OR«-Funktion mit der Zahl 64 verknüpft. In Tabelle 2 sehen Sie an dem Beispiel »PRG«, wie die OR-Verknüpfung arbeitet. Die beiden Operanten werden Bit für Bit verglichen. Ist in einem der Faktoren ein Bit gesetzt, also 1, wird es auch im Ergebnisbyte gesetzt. Um das sechste Bit wieder auf 0 zu setzen, wählen wir die Verknüpfung »AND«. Jetzt ergibt sich eine 1 im Ergebnisbyte nur dann, wenn beide Faktoren den Wert 1 haben. Diese Eigenschaft können wir nutzen, um eine bestimmte Bitstelle auf 0 zu setzen. Wir verknüpfen das geschützte Byte mit dem Faktor 191 (dual 1011 1111) und erhalten wieder den ungeschützten Filetyp 130 (Tabelle 3).

Genug der grauen Theorie! Wie kann man diese Änderungen nun auf der Diskette vornehmen? Dazu müssen wir den Filetyp (1 Byte) von der Disk lesen, ihn ändern und dann wieder zurückschreiben. Das Betriebssystem der 1541 sieht hierzu die Befehle »U1« (Block lesen) und »U2« (Block schreiben) vor. Mit diesen Befehlen können Sie aber nur ganze Blöcke, also 256 Byte in den Puffer einer Direktzugriffsdatei einlesen oder aus ihm auf die Diskette schreiben. Da wir nur ein bestimmtes Byte benötigen, nutzen wir die Möglichkeit, den Zeiger, der die augenblickliche Lese- oder Schreibposition im Puffer angibt, auf einzelne Bytes setzen zu können. Der Befehl hierfür lautet »B-P« (Buffer Pointer). Alle Befehle, die für die Floppy bestimmt sind, werden über den Befehlskanal gesendet, der im »OPEN«-Befehl (Zeile 640) durch die Sekundäradresse 15 angesprochen wird. Mit dem PRINT #-Befehl werden jetzt die Kommandos übermittelt. Zum Beispiel lesen eines Sektors von Spur 18 (Zeile 2160). Daten können nicht direkt von der Diskette in den Computer eingelesen werden, sondern müssen in einem Datenpuffer zwischengespeichert werden. Dazu eröffnen wir eine Direktzugriffsdatei (Zeile 650), aus der wir dann mit dem GET #-Befehl die Bytes einzeln herauslesen.

Im Bild 7 sehen Sie ein kleines Beispiel für die Anwendung der genannten Befehle. Mit diesem Unterprogramm können Sie den Namen einer Diskette einlesen.

Zeile	Kommentar
100-350	Titel des PGMs, Adresse des Autors
360-450	Haupt-Programm; von hier aus werden die einzelnen Unterprogramme angesprungen. Aus welchen Zeilen die Unterprogramme angesprungen werden, entnehmen Sie bitte Tabelle 4.
460-530	Setzen der Bildschirmfarben und festlegen der Konstanten »BL$«
540-600	Unter-PGM warten auf Tastendruck und ein Zeichen aus dem Tastaturpuffer holen.
610-660	Unter-PGM: Befehlskanal und Direktzugriffsdatei auf der Floppy öffnen.
670-730	Unter-PGM: Zeichen aus Floppy-Puffer einlesen und sicherstellen, daß kein Leerstring weitergegeben wird.
740-960	Ausgabe des Titelbildes und Menüs
970-1250	In diesem Unterprogramm wird geprüft, ob eine Disk eingelegt ist oder ob bei Protect und Unprotect ein Schreibschutz auf der Disk vorhanden ist. Außerdem wird wenn kein Fehler auftrat, bei Protect und Unprotect eine Tabellenüberschrift ausgegeben.
1260-1430	Unter-PGM. Filetyp einlesen und auswerten.
1440-2040	Unter-PGM. Directory einlesen.
2050-2510	Unter-PGM: Protect oder Unprotect. Ist im Haupt-PGM P=1 dann wird die Routine Protect, ist P=0 die Routine Unprotect durchgeführt.

Die Funktionen der einzelnen Variablen entnehmen Sie bitte Tabelle 5. Eine wichtige Anmerkung: Damit Fileprotect 64 einwandfrei arbeitet, muß in Zeile 2499 hinter dem »(FT)« unbedingt ein Semikolon (;) gesetzt sein! Ist dies nicht der Fall, können Sie geschützte Programme nicht mehr laden. Ohne Semikolon wird ein CHR$(13) an den Wert angehängt. Es würden also zwei Byte und damit die Startadresse des Programms überschrieben!

(Jochen Fette / rg)

ABB. : 1

```
FILEPROTECT 64

SCRATCH-SCHUTZ VON FILES FUER
C-64 UND 1541

1984  BY JOCHEN FETTE

F1   DIRECTORY

F3   PROTECT

F5   UNPROTECT

F8   END

     IHRE WAHL ?
```

ABB. : 2

```
DIRECTORY

DISK EINLEGEN UND TASTE DRUECKEN

DISK FEHLER !
```

ABB. : 3

```
DIRECTORY

DISK EINLEGEN UND TASTE DRUECKEN .

 2   TEST 1              PRG
 1   DISK ERROR          PRG
 5   DRUCK MASKE2        PRG
 1   DRUCK$ MASKE        PRG
 5   DRUCK TEST1         PRG
 3   GUNIS KARTEI-KAR    PRG
 115 ↑ ART.              SEQ
 9   TEST 1              PRG
 33  3-D 1G              PRG
 33  3-D 2G              PRG<
 4   TEST                PRG<
 12  FILEPROTECT 64      PRG
 3   BALKEN              PRG

 438    FREIE BLOECKE

TASTE DRUECKEN . . .
```

ABB. : 4

```
PROTECT

DISK EINLEGEN UND TASTE DRUECKEN .

SCHREIBSCHUTZ ENTFERNEN !
```

ABB. : 5

```
PROTECT

DISK EINLEGEN UND TASTE DRUECKEN . . .

 TYP  NAME
 ----------------------------------------------
 PRG  TEST 1              PROTECT (J/N)N
 PRG  DISK ERROR          PROTECT (J/N)N
 PRG  DRUCK MASKE2        PROTECT (J/N)J
 PRG  DRUCK$ MASKE        PROTECT (J/N)J
 PRG  DRUCK TEST1         PROTECT (J/N)N
 PRG  GUNIS KARTEI-KAR    PROTECT (J/N)N
 SEQ  ↑ ART.              PROTECT (J/N)N
 PRG  TEST 1              PROTECT (J/N)J
 PRG  3-D 1G              PROTECT (J/N)J
 PRG  FILEPROTECT 64      PROTECT (J/N)J
 PRG  BALKEN              PROTECT (J/N)N
```

ABB. : 6

```
UNPROTECT

DISK EINLEGEN UND TASTE DRUECKEN . . .

 TYP  NAME
 ----------------------------------------------
 PRG< DRUCK MASKE2       UNPROTECT (J/N)N
 PRG< DRUCK$ MASKE       UNPROTECT (J/N)N
 PRG< TEST 1             UNPROTECT (J/N)J
 PRG< 3-D 1G             UNPROTECT (J/N)J
 PRG< 3-D 2G             UNPROTECT (J/N)J
 PRG< TEST               UNPROTECT (J/N)J
 PRG< FILEPROTECT 64     UNPROTECT (J/N)J
```

ABB. : 7

```
100 OPEN15,8,15,"I0"
110 OPEN2,8,2,"#"
120 PRINT#15,"B-R";2;0;18;0
130 PRINT#15,"B-P";2;144
140 N$=""
150 FOR I=0 TO 15
160 GET#2,X$:IF X$=CHR$(160) THEN I=15:GOTO 180
170 N$=N$+X$
180 NEXT I
190 PRINT "DISK NAME: ";N$
200 CLOSE 2
210 CLOSE 15
220 END

READY.
```

TABELLE 1

Dual	Dez	Hex	Filetyp
0000 0000	..0	$00	*DEL
0000 0001	..1	$01	*SEQ
0000 0010	..2	$02	*PRG
0000 0011	..3	$03	*USR
0000 0100	..4	$04	*REL
1000 0000	128	$80	DEL
1000 0001	129	$81	SEQ
1000 0010	130	$82	PRG
1000 0011	131	$83	USR
1000 0100	132	$84	REL
1100 0000	192	$C0	DEL<
1100 0001	193	$C1	SEQ<
1100 0010	194	$C2	PRG<
1100 0011	195	$C3	USR<
1100 0100	196	$C4	REL<

TABELLE 2

Die 'OR - Verknuepfung'

```
Dual        Dez   Typ
-------------------------------------
1000 0010   130   PRG
0100 0000   .64   Verknuepfungsfaktor
===================
1100 0010   194   PRG<
```

TABELLE 3

Die 'AND - Verknuepfung'

```
Dual       Dez  Typ
-----------------------------------
1100 0010  194  PRG<
1011 1111  191  Verknuepfungsfaktor
===================
1000 0010  130  PRG
```

TABELLE 5

```
A$  ->  Get von Tastatur
BB  ->  Belegte Blocks
BL$ ->  Formatstring fuer Bildschirmausgaben
FB  ->  Freie Blocks
FT$ ->  Filetyp als Wort
FT  ->  Filetyp als Zahl
HB  ->  High Byte der belegten Blocks
I   ->  Laufvariable
LB  ->  Low Byte der belegten Blocks
N$  ->  Name des aktuellen Directoryeintrages
P   ->  Schalter fuer Protect oder Unprotect
RM  ->  Rueckmeldung von Unterprogrammen
S   ->  Aktueller Sektor
SA  ->  Alter Sektor
T   ->  Track
UE  ->  Schalter fuer Ueberschrift ja/nein
X$  ->  Get von Diskette
Y   ->  Laufvariable
ZE  ->  Zeilenzaehler bei Unter-PGM Directory
```

TABELLE 4

S P R U N G T A B E L L E

```
Zeile  Wird angesprungen von Zeile
----------------------------------
 370    440*
 380    400?
 490    360@
 570    380@   1020@   1890@   2030@   2420@
 640   1070@   1530@   2140@
 700   1290@   1590@   1610@   1680@   1700@
       1780@   2170@   2190@   2330@   2360@
 770    370@
1000   1510@   2120@          *  = GOTO
1140   1060?                  @  = GOSUB
1180   1140?                  ?  = IF THEN
1250   1180?   1210?*         ?* = IF THEN GOTO
1290   1650@   2240@          ?@ = IF THEN GOSUB
1470    430@   1520?
1570   1940?
1810   1790?*
1930   1660?   1840?
2080    430@   2130?
2160   2290?
2330   2250?@  2260?@
2420   2440?
2510   2430?*
```

```
100 REM ----------------------------------------------------------------  <110>
110 :                                                                     <168>
120 REM F I L E P R O T E C T  6 4                                        <178>
130 :                                                                     <188>
140 REM SCRATCH-SCHUTZ VON FILES                                          <151>
150 :                                                                     <208>
160 REM         FUER                                                      <097>
170 :                                                                     <228>
180 REM     C-64 UND 1541                                                 <207>
190 :                                                                     <248>
200 REM ..............................................................    <025>
210 :                                                                     <012>
220 REM     JOCHEN  FETTE                                                 <154>
230 :                                                                     <032>
240 REM    GASSELSTIEGE 14                                                <100>
250 :                                                                     <052>
260 REM       4400 MUENSTER                                               <207>
```

```
270 :                                                                    <073>
280 :                                                                    <083>
290 REM ---------------------------------------------------------------- <045>
300 :                                                                    <103>
310 :                                                                    <113>
320 :                                                                    <123>
330 :                                                                    <133>
340 :                                                                    <143>
350 :                                                                    <153>
360 GOSUB 490:REM                                      INITIALISIEREN    <117>
370 GOSUB 770:REM                                          TITELBILD     <004>
380 GOSUB 570:REM                                   WARTEN AUF TASTE     <157>
390 IF ASC(A$)=140 THEN SYS 64738:REM                   ENDE? -> RESET!
                                          <089>
400 IF ASC(A$)>135 OR ASC(A$)<133 THEN 380:REM          UNGUELTIGE TASTE
                                          <221>
410 IF ASC(A$)=134 THEN P=1:REM                                  PROTECT <176>
420 IF ASC(A$)=135 THEN P=0:REM                                UNPROTECT <093>
430 ON ASC(A$)-132 GOSUB 1470,2080,2080                                  <066>
440 GOTO 370                                                             <220>
450 END                                                                  <067>
460 REM --------------------------------------------------------------- <170>
470 REM INITIALISIEREN                                                   <127>
480 REM --------------------------------------------------------------- <190>
490 BL$="[SPACE20]":REM                          20 - SPACE              <087>
500 POKE 53280,9:REM                    FARBE FUER RAND AUF 'BRAUN'      <021>
510 POKE 53281,9:REM              FARBE FUER HINTERGRUND AUF 'BRAUN'     <069>
520 PRINT CHR$(5);:REM                FARBE FUER SCHRIFT AUF 'WEISS'     <078>
530 RETURN                                                               <162>
540 REM --------------------------------------------------------------- <251>
550 REM WARTEN AUF BELIEBIBE TASTE UND EIN ZEICHEN VON TASTATUR HOLEN    <002>
560 REM --------------------------------------------------------------- <015>
570 A$=""                                                                <151>
580 POKE 198,0:WAIT 198,255:REM              WARTEN AUF BELIEBIGE TASTE  <128>
590 GET A$                                                               <086>
600 RETURN                                                               <232>
610 REM ---------------------------------------------------------------- <065>
620 REM DISK OEFFNEN                                                     <041>
630 REM ---------------------------------------------------------------- <085>
640 OPEN 15,8,15,"I0":REM       BEFEHLSKANAL 15 ,UND 'I0' FUER INITIALISIEREN  <026>
650 OPEN 2,8,2,"#":REM          DATEN PUFFER 2  ,UND '#'  FUER DIRECKTZUGRIFF  <186>
660 RETURN                                                               <036>
670 REM ---------------------------------------------------------------- <125>
680 REM 1 ZEICHEN AUS DISK LESEN                                         <251>
690 REM ---------------------------------------------------------------- <145>
700 X$=""                                                                <048>
710 GET#2,X$:REM                                       1 ZEICHEN LESEN   <221>
720 IF X$=""THEN X$=CHR$(0):REM         SONST 'ILLEGAL QUANTITY ERROR'   <157>
730 RETURN                                                               <106>
740 REM ---------------------------------------------------------------- <195>
750 REM TITELBILD AUSGEBEN                                               <102>
760 REM ---------------------------------------------------------------- <215>
770 PRINT CHR$(147);:REM                                          'CLR'  <133>
780 PRINT SPC(13);"FILEPROTECT[SPACE]64"                                 <005>
790 PRINT SPC(12);"EEEEEEEEEEEEEEEE":REM                   'SCHIFT + E'  <251>
800 PRINT                                                                <156>
810 PRINT SPC(6);CHR$(18);"SCRATCH-SCHUTZ[SPACE]VON[SPACE]FILES[SPACE]FUER";CHR$(146)
  :REM   INVERS                                                          <066>
820 PRINT SPC(13);CHR$(18);"C-64[SPACE]UND[SPACE2]1541";CHR$(146)
  :REM              INVERS                                               <088>
830 PRINT                                                                <218>
840 PRINT CHR$(144);"[SPACE8](C)[SPACE]1984[SPACE2]BY[SPACE]JOCHEN[SPACE]FETTE";CHR$(5)
  :REM      SCHWARZ                                                      <187>
```

```
850 PRINT:PRINT:PRINT:PRINT                                                      <103>
860 PRINT SPC(10);CHR$(18);"[SPACE]F1[SPACE]";CHR$(146);"[SPACE3]DIRECTORY"     <252>
870 PRINT                                                                        <002>
880 PRINT SPC(10);CHR$(18);"[SPACE]F3[SPACE]";CHR$(146);"[SPACE3]PROTECT"       <126>
890 PRINT                                                                        <022>
900 PRINT SPC(10);CHR$(18);"[SPACE]F5[SPACE]";CHR$(146);"[SPACE3]UNPROTECT"     <055>
910 PRINT                                                                        <042>
920 PRINT SPC(10);CHR$(18);"[SPACE]F8[SPACE]";CHR$(146);"[SPACE3]END"           <097>
930 PRINT                                                                        <062>
940 PRINT                                                                        <072>
950 PRINT SPC(10);"[SPACE7]IHRE[SPACE]WAHL[SPACE]?";:REM  7 - SPACE             <104>
960 RETURN                                                                       <081>
970 REM ---------------------------------------------------------------------   <170>
980 REM DISK EINGELEGT ? KEIN SCHREIBSCHUTZ ?                                    <171>
990 REM ---------------------------------------------------------------------   <190>
1000 PRINT                                                                       <132>
1010 PRINT"DISK[SPACE]EINLEGEN[SPACE]UND[SPACE]TASTE[SPACE]DRUECKEN[SPACE].[SPACE].
  [SPACE]."                                                                      <135>
1020 GOSUB 570:REM                                            WARTEN AUF TASTE   <031>
1030 OPEN 15,8,15,"I0":REM       OEFFNEN DES BEFEHLSKANALS UND INITIALISIEREN   <071>
1040 INPUT#15,RM:REM                                               FEHLER CODE   <099>
1050 CLOSE 15                                                                    <036>
1060 IF RM<>0 THEN 1140:REM                                         KEINE DISK   <179>
1070 GOSUB 640:REM                                                 DISK OEFFNEN  <078>
1080 PRINT#15,"B-R";2;0;18;0:REM                            LESEN EINES BLOCKS   <196>
1090 PRINT#15,"B-W";2;0;18;0:REM                        SCHREIBEN DES BLOCKS    <087>
1100 INPUT#15,RM:REM                                               FEHLER CODE   <159>
1110 CLOSE 15:REM                                  BEFEHLSKANAL SCHLIESSEN      <122>
1120 CLOSE 2:REM                                    DATENPUFFER SCHLIESSEN      <036>
1130 PRINT                                                                       <007>
1140 IF RM<>0 OR UE=0 THEN 1180:REM    FEHLER,AUFRUF VON DIRECTORY -> KEINE UEBER.
                                   <103>
1150 PRINT"[SPACE]TYP[SPACE2]NAME"                                               <125>
1160 PRINT"EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE":REM        39 - SCHIFT 'E'   <034>
1170 UE=0                                                                        <018>
1180 IF RM=0 THEN 1250:REM                               KEIN FEHLER -> RETURN  <012>
1190 PRINT                                                                       <067>
1200 IF RM<>26 THEN PRINT CHR$(144);"DISK[SPACE]FEHLER[SPACE]!";CHR$(5)
  :REM     SCHWARZE SCHRIFT                                                      <231>
1210 IF UE=0 AND RM=26 THEN RM=0:GOTO 1250:                                      <026>
1220 IF RM=26 THEN PRINT CHR$(144);"SCHREIBSCHUTZ[SPACE]ENTFERNEN[SPACE]!";CHR$(5):
                                   <229>
1230 FOR W=1 TO 1500:REM                                          WARTESCHLEIFE
                                   <134>

1240 NEXT W                                                                      <181>
1250 RETURN                                                                      <116>
1260 REM --------------------------------------------------------------------    <205>
1270 REM FILETYP HOLEN                                                           <028>
1280 REM --------------------------------------------------------------------    <226>
1290 GOSUB 700:REM                                      1 ZEICHEN AUS PUFFER    <228>
1300 FT=0:FT$=""                                                                 <131>
1310 IF ASC(X$)=1 THEN FT=1:FT$="*SEQ[SPACE]"                                    <073>
1320 IF ASC(X$)=2 THEN FT=2:FT$="*PRG[SPACE]"                                    <085>
1330 IF ASC(X$)=3 THEN FT=3:FT$="*USR[SPACE]"                                    <114>
1340 IF ASC(X$)=4 THEN FT=4:FT$="*REL[SPACE]"                                    <103>
1350 IF ASC(X$)=129 THEN FT=129:FT$="[SPACE]SEQ[SPACE]"                          <029>
1360 IF ASC(X$)=130 THEN FT=130:FT$="[SPACE]PRG[SPACE]"                          <023>
1370 IF ASC(X$)=131 THEN FT=131:FT$="[SPACE]USR[SPACE]"                          <052>
1380 IF ASC(X$)=132 THEN FT=132:FT$="[SPACE]REL[SPACE]"                          <041>
1390 IF ASC(X$)=193 THEN FT=193:FT$="[SPACE]SEQ<"                                <131>
1400 IF ASC(X$)=194 THEN FT=194:FT$="[SPACE]PRG<"                                <143>
1410 IF ASC(X$)=195 THEN FT=195:FT$="[SPACE]USR<"                                <172>
```

```
1420 IF ASC(X$)=196 THEN FT=196:FT$="[SPACE]REL<"                          <161>
1430 RETURN                                                                 <041>
1440 REM ---------------------------------------------------------------    <130>
1450 REM  DIRECTORY EINLESEN                                                <070>
1460 REM ---------------------------------------------------------------    <150>
1470 PRINT CHR$(147):REM                                           'CLR'   <008>
1480 PRINT SPC(6);CHR$(18);"D[SPACE]I[SPACE]R[SPACE]E[SPACE]C[SPACE]T[SPACE]O[SPACE]R
  [SPACE]Y";CHR$(146):REM              REVERS                              <233>
1490 PRINT                                                                  <112>
1500 UE=0:REM                                         KEINE UEBERSCHRIFT   <024>
1510 GOSUB 1000:REM                                   DISK EINGELEGT?      <000>
1520 IF RM<>0 THEN 1470:REM                             NEUER VERSUCH      <142>
1530 GOSUB 640:REM                                      DISK OEFFNEN       <027>
1540 T=18:S=1:REM                                     TRACK UND SEKTOR     <230>
1550 BB=0:FB=664:REM                       BELEGTE UND UNBELEGTE BLOECKE   <198>
1560 ZE=0:REM                          ZAEHLER FUER BILDSCHIERMZEILEN      <146>
1570 PRINT#15,"B-R";2;0;T;S:REM                        1. SEKTOR LESEN     <195>
1580 PRINT#15,"B-P";2;0:REM               POINTER AUF NAECHSTEN TRACK      <011>
1590 GOSUB 700:REM                                     ZEICHEN LESEN       <166>
1600 T=ASC(X$):REM                                   NAECHSTER TRACK       <186>
1610 GOSUB 700                                                              <116>
1620 S=ASC(X$):REM                                   NAECHSTER SEKTOR      <048>
1630 FOR I=0 TO 7:REM                   8 DIRECKTORY EINTRAEGE PRO SEKTOR
                                         <073>
1640 PRINT#15,"B-P";2;I*32+2:REM                  POINTER AUF FILETYP     <092>
1650 GOSUB 1290:REM                                    FILETYP HOLEN       <045>
1660 IF FT=0 THEN 1930:REM               NAECHSTER DIRECTORY EINTRAG
                                         <034>
1670 PRINT#15,"B-P";2;I*32+30:REM        POINTER AUF ANZAHL DER BLOCKS     <229>
1680 GOSUB 700                                                              <186>
1690 LB=ASC(X$):REM                                  LOW BYTE DEZIMAL      <104>
1700 GOSUB 700                                                              <206>
1710 HB=ASC(X$)*256:REM                             HIGH BYTE DEZIMAL      <239>
1720 BB=HB+LB:REM                                     BELEGTE BLOCKS       <053>
1730 PRINT STR$(BB)+LEFT$(BL$,5-LEN(STR$(BB)));:REM  AUSGABE DER BLOCKS    <189>
1740 FB=FB-BB:REM                                       FREIE BLOCKS       <181>
1750 PRINT#15,"B-P";2;I*32+5:REM                   POINTER AUF FILE NAME   <241>
1760 N$="":REM                              ALTEN FILENAMEN LOESCHEN      <107>
1770 FOR Y=0 TO 15:REM                                 MAXIMAL 16 ZEICHEN
                                         <245>
1780 GOSUB 700                                                              <030>
1790 IF ASC(X$)=160 THEN Y=15:GOTO 1810:REM   SCHIFT SPACE NICHT UEBERNEHMEN
                                         <170>
1800 N$=N$+X$                                                               <203>
1810 NEXT Y                                                                 <244>
1820 PRINT N$+LEFT$(BL$,20-LEN(N$));FT$:REM         FILENAMEN UND FILETYP  <183>
1830 ZE=ZE+1:REM                                        ZEILENZAEHLER      <147>

1840 IF ZE<15 THEN 1930:REM                         SEITE NOCH NICHT VOLL
                                         <012>
1850 ZE=0                                                                   <194>
1860 PRINT                                                                  <228>
1870 PRINT"WEITERE[SPACE]EINTRAEGE"                                         <219>
1880 PRINT"BITTE[SPACE]TASTE[SPACE]DRUECKEN[SPACE].[SPACE].[SPACE].";      <075>
1890 GOSUB 570:REM                                    WARTEN AUF TASTE     <137>
1900 PRINT CHR$(147):REM                                           'CLR'   <184>
1910 PRINT SPC(6);CHR$(18);"D[SPACE]I[SPACE]R[SPACE]E[SPACE]C[SPACE]T[SPACE]O[SPACE]R
  [SPACE]Y";CHR$(146):REM              INVERS                              <153>
1920 PRINT:PRINT:PRINT:PRINT                                                <153>
1930 NEXT I:REM                              NAECHSTER DIRECTORY EINTRAG  <129>
1940 IF T<>0 THEN 1570:REM                          WEITERE EINTRAEGE ?
                                         <051>
1950 CLOSE 2:REM                                     PUFFER SCHLIESSEN     <249>
```

```
1960 GLOSE 15:REM                                  BEFEHLSKANAL SCHLIESSEN  <207>
1970 PRINT                                                                  <082>
1980 PRINT FB;:REM                                        FREIE BLOECKE     <072>
1990 IF FB<>1 THEN PRINT"[SPACE3]FREIE[SPACE]BLOECKE"                       <089>
2000 IF FB=1 THEN PRINT"[SPACE3]FREIER[SPACE]BLOCK"                         <121>
2010 PRINT                                                                  <122>
2020 PRINT"TASTE[SPACE]DRUECKEN[SPACE].[SPACE].[SPACE].";                   <095>
2030 GOSUB 570:REM                                   WARTEN AUF TASTE       <021>
2040 RETURN                                                                 <141>
2050 REM  --------------------------------------------------------------    <231>
2060 REM  PROTECT ODER UNPROTECT                                            <178>
2070 REM  --------------------------------------------------------------    <251>
2080 PRINT CHR$(147):REM                                          'CLR'     <077>
2090 IF P=1 THEN PRINT SPC(13);CHR$(18);"P[SPACE]R[SPACE]O[SPACE]T[SPACE]E[SPACE]C
   [SPACE]T";CHR$(146):REM        INVERS                                    <077>
2100 IF P=0 THEN PRINT SPC(12);CHR$(18);"U[SPACE]N[SPACE]P[SPACE]R[SPACE]O[SPACE]T
   [SPACE]E[SPACE]C[SPACE]T";CHR$(146)                                      <088>
2110 UE=1:REM                                             UEBERSCHRIFT      <018>
2120 GOSUB 1000:REM                                      DISK EINGELEGT?    <101>
2130 IF RM<>0 THEN 2080:REM                  WENN FEHLER DANN NEUER VERSUCH
                                     <000>
2140 GOSUB 640:REM                                        DISK OEFFNEN      <128>
2150 T=18:S=1:REM                                      TRACK UND SEKTOR     <074>
2160 PRINT#15,"U1[SPACE]2[SPACE]0[SPACE]18",S:REM      SEKTOR 'S' VON TRACK 18 IN PUFFE
   R '2' LESEN                                                              <031>
2170 GOSUB 700:REM                                        ZEICHEN LESEN     <236>
2180 T=ASC(X$):REM                                      NAECHSTER TRACK     <000>
2190 GOSUB 700                                                              <186>
2200 SA=S:REM                                  AKTUELLEN SEKTOR MERKEN      <065>
2210 S=ASC(X$):REM                                     NAECHSTER SEKTOR     <128>
2220 FOR I=2 TO 226 STEP 32                                                 <174>
2230 PRINT#15,"B-P";2;I:REM                          POINTER AUF FILETYP    <191>
2240 GOSUB 1290:REM                                        FILETYPHOLEN     <125>
2250 IF FT>128 AND FT<133 AND P=1 THEN GOSUB 2330:REM NUR UNGESCHUETZTE FILES FUER PROT.
                                     <120>
2260 IF FT>192 AND FT<197 AND P=0 THEN GOSUB 2330:REM NUR GESCHUETZTE FILES FUER UNPROT.
                                     <140>
2270 NEXT I:REM                              NAECHSTER DIRECTORY EINTRAG    <214>
2280 IF RM=1 THEN RM=0:PRINT#15,"U2[SPACE]2[SPACE]0[SPACE]18";SA
   :REM       KORRIGIRTEN BLOCK SCHREIBEN                                   <005>
2290 IF T<>0 THEN 2160:REM                                 WEITERE EINTRAEGE
                                     <079>
2300 CLOSE 2:REM                                  DATENPUFFER SCHLIESSEN    <196>
2310 CLOSE 15:REM                                BEFEHLSKANAL SCHLIESSEN    <047>
2320 RETURN:REM                                          DIRECTORY ENDE     <065>
2330 GOSUB 700:GOSUB 700:REM                       2BYTE UEBERSPRINGEN      <173>
2340 N$="":REM                                    ALTEN NAMEN LOESCHEN      <146>
2350 FOR Y=0 TO 15:REM                              HOECHSTENS 16 ZEICHEN
                                     <039>
2360 GOSUB 700                                                              <101>
2370 IF ASC(X$)=0 THEN N$=N$+"[SPACE]":REM         IMMER AUFFUELLEN AUF 16 ZEICH
   EN                                                                       <233>
2380 IF ASC(X$)<>0 THEN N$=N$+X$                                            <106>
2390 NEXT Y                                                                 <058>
2400 IF P=1 THEN PRINT FT$+"[SPACE]"+N$+"[SPACE3]PROTECT[SPACE](J/N)";
   :REM           FILETYP UND NAME                                          <127>
2410 IF P=0 THEN PRINT FT$+"[SPACE]"+N$+"[SPACE]UNPROTECT[SPACE](J/N)";
   :REM           FILETYP UND NAME                                          <043>
2420 GOSUB 570:REM                                   WARTEN AUF TASTE       <157>
2430 IF A$="N"THEN PRINT"N":GOTO 2510:REM                            RETURN
                                     <193>
2440 IF A$<>"J"THEN 2420:REM                             UNGUELTIGE TASTE <037>
2450 PRINT"J"                                                               <194>
```

```
2460 PRINT#15,"B-P";2;I:REM                              PUFFER POINTER AUF FILETYP  <110>
2470 IF P=1 THEN FT=(FT OR 64):REM                                  FILETYP SCHUETZEN
                                                        <004>
2480 IF P=0 THEN FT=(FT AND 255-64):REM                       SCHUTZ ZURUECKNEHMEN
                                                        <066>
2490 PRINT#2,CHR$(FT);:REM                                 GEAENDERTER FILETYP _ <178>
2500 RM=1:REM                                       VERAENDERUNGEN DURCHGEFUEHRT <001>
2510 RETURN                                                                      <101>
```

P-Basic-V2: Autostart mit Rückwärtsgang

Dieser Autostarter erzeugt nicht nur ein einfaches Lade-Programm für Basic und Assembler-Programme. Es wird zusätzlich die RUN/STOP-Taste gesperrt und ein Kopierschutz installiert. Um ein so behandeltes Programm wieder verändern zu können, kann mit einem Befehl der gesamte Vorgang rückgängig gemacht werden.

Das Programm »P-Basic-V2« läuft auf dem Commodore 64 (ohne Basic-Erweiterung) und der Floppy 1541 (nicht auf der Datasette, da das Programm beim Laden den Kassettenpuffer benutzt). Es ist zirka 260 Bytes lang und liegt im Speicher ab 49152 ($c000).

Nach dem normalen Laden und Starten des Basic-Programmes, ist das Basic-V2 um drei neue Befehle erweitert:

1. !SAVE"Name", Gerätenummer,1
2. !LOAD"Name", Gerätenummer,1
3. !NEW

Der Hauptbefehl ist !SAVE. Mit ihm kann man ein Basic-Programm oder ein Maschinenprogramm (mit Basic-Startzeile) so abspeichern, daß es nach dem Laden von allein startet. Außerdem wird die RUN/STOP/RESTORE-Taste abgestellt. Das Programm ist somit vor Änderungen gesichert. Auch das Kopieren ist nur mit besonderen Kopierprogrammen möglich. Doch durch eine Codewortabfrage in den ersten Zeilen wird es wohl keiner schaffen, das Programm zum Laufen zu bringen. Ein Programm, das mit dieser Basic-Hilfe abgespeichert wurde, muß mit dem Zusatz »,1« absolut, nicht ab Basic-Start (2049), geladen werden. Lädt man ein solches Programm ohne diesen kleinen Zusatz in den Speicher, so läßt es sich weder listen, noch starten. Weil ein normales Programm mit der Zwei-Byte-Startadresse am Anfang der Datei gespeichert wird, ist es durch ein nochmaliges anderes Abspeichern ganz verloren.

Man kann diese Routine natürlich auch nur zur Bequemlichkeit benutzen. Um die Stopp-Routine wieder zu aktivieren, hilft: POKE808,237 als erster Befehl des Programms.

Ist in dem geschützten Programm nun aber ein Fehler, oder will man es erweitern oder verbessern, so hilft der zweite Befehl. Mit !LOAD kann man alle Programme, die mit Autostart geschützt sind, knacken und nach eigener Lust bearbeiten.

Tritt bei den Arbeiten mit diesen Befehlen ein Fehler auf, den Sie oder das Betriebssystem (Floppy/Computer) verursacht haben, so können sie mit Eingabe von !NEW alle wichtigen Vektoren korrigieren und somit das gerade bearbeitete Programm wieder listen lassen.

Erläuterungen zum Assemblerlisting

In dem Programmabschnitt »Vektorsetzung« wird die Routine zur Erkennung der Befehls-Token (Basic-Befehle) auf die erweiterte Routine (Erkennung) umgelenkt und danach der Name der Basic-Erweiterung ausgegeben.

In der Erkennungsroutine werden die Befehls-Token mit dem neuen Befehl (!) und danach mit den eigentlichen Befehlen (SAVE, LOAD, NEW) verglichen. »SAVE« hat das Basic-Token 148, »LOAD« hat 147 und »NEW« hat 162. Wird keiner dieser Befehle interpretiert, so wird Syntax Error (Jmp $af08) angezeigt. Die DATAs werden in der Vektorsetzungsroutine ausgegeben.

Der Befehl »LOAD«, der wie der Befehl »SAVE« nur im Direktmodus verwendet werden darf, setzt zuerst das Programmodusflag, damit es keine Ausgabe wie »searching for ...« und so weiter gibt. Danach werden die LOADparameter geholt ($e1d4) und nach LOAD verzweigt ($ffd5). Nachdem das Programm geladen ist, werden alle Vektoren mit der NEW-Routine wieder zurückgesetzt und dann zur END-Routine verzweigt ($a831).

Der Befehl »SAVE« gleicht am Anfang (bis Ende Zeile 166) dem LOAD-Befehl. In Zeile 168 wird die Steuerroutine in den Kassettenpuffer (Sprite 11) geladen. Dann werden die »Input«- und »Output«-Vektoren auf die Steuerroutine gesetzt und das Programm ab Byte 806 gespeichert. Die Vektoren werden dann mit der NEW-Routine normalisiert.

Die Steuerroutine arbeitet ungefähr so wie die NEW-Routine. Nur bei dieser Routine werden zusätzlich die RUN-Zeiger ($a659) gesetzt, die Run-Stop/Restore-Funktion abgeschaltet und dann zur Interpreterschleife verzweigt.

Bemerkungen zum Programm

Das Befehlszeichen »!« kann natürlich durch fast jedes Zeichen ersetzt werden, nur wenn das Zeichen ein Token ist, muß nicht der ASCII-Wert, sondern der Tokenwert angegeben werden, da sonst der Befehl nicht erkannt wird. Wenn das Zeichen ein Buchstabe ist, kann derselben Variablen kein Wert zugewiesen werden. Denn auch dabei entsteht ein Fehler.

(Jan Kusch/gk)

Markt&Technik Verlag Aktiengesellschaft Buchverlag

G. Tatzl

Praktische Problemanalyse

1983, 319 Seiten
Dieses Buch — aus den Erfahrungen langjähriger Praxis für die Praxis geschrieben — leistet einen Beitrag zur Behebung der vielzitierten Softwarekrise. Ohne den Leser in ein enges Denkschema zu pressen, wird bei Wahrung eines Mindestmaßes an Systematik vorwiegend die kreative und intuitive Seite angesprochen. Anhand einfacher bis mittlerer Beispiele wird in diesem Buch der Weg eines »springenden Funkens« verfolgt. Hieraus lassen sich optimale Problemgestaltungen ableiten.

Bestellnummer VO 612 DM 22,50

G. Oetzmann

Programmieren von Mikrocomputern 2

1981, 115 Seiten, mit zahlreichen Beispielen, 8 vollständigen Programmen und 32 Bildern
Die Rechner der Serien cbm 2001 und cbm 3001 können in cbm-BASIC programmiert werden, einer BASIC-Variante mit verschiedenen herstellerspezifischen Besonderheiten. Dieser zweite Band der Reihe »Programmieren von Mikrocomputern« enthält eine umfassende Einführung in cbm-BASIC, ohne daß Programmierkenntnisse vorausgesetzt werden. Neben der Vermittlung der Sprache wird an verschiedenen Beispielen der Weg von der ersten Lösungsidee bis zum fertigen Programm demonstriert.

Bestellnummer VV 313 DM 14,90

G. Oetzmann

Programmieren von Mikrocomputern 5

Lehr- und Übungsbuch für die Rechnerserien CBM 4001 und CBM 8001
1983, 119 Seiten
Die Rechner der Serien CBM 4001 und CBM 8001 können in CBM-Basic programmiert werden, einer Basic-Variante mit verschiedenen herstellerspezifischen Besonderheiten. Dieser Band enthält eine umfassende Einführung in CBM-Basic, ohne daß Programmierkenntnisse vorausgesetzt werden. Neben der Vermittlung der Sprache wird an verschiedenen Beispielen der Weg von der ersten Lösungsidee bis zum fertigen Programm demonstriert.

Bestellnummer VV 346 DM 14,90

G. Oetzmann

Programmieren von Mikrocomputern 7

Lehr- und Übungsbuch für Commodore-Volkscomputer
1983, 109 Seiten
Commodore-Volkscomputer können in einer gegenüber der Standardversion deutlich erweiterten Variante der Programmiersprache Basic programmiert werden. Dieses Buch bietet eine umfassende Einführung in VC-Basic, ohne daß Programmierkenntnisse vorausgesetzt werden. Besonderer Wert wurde darauf gelegt, neben der Vermittlung der Sprache auch den Weg von der ersten Lösungsidee bis zum fertigen Programm an verschiedenen Beispielen zu demonstrieren.

Bestellnummer VV 576 DM 12,40

P. Kahlig

Programmieren von Mikrocomputern 8

Assembler-Programmierung von Mikroprozessoren (8080, 8085, Z80) mit dem ZX81
1983, 185 Seiten
Das Buch bietet Übungen zur Assembler-Programmierung der Mikroprozessoren 8080, 8085 und Z80, die mit dem Mikrocomputer Sinclair ZX 81 durchgeführt wird. Der Leser soll dadurch angeregt werden, sich vertiefte Hardware- und Software-Kenntnisse anzueignen.
Aus dem Inhalt: Hilfsprogramm »KEY« — Eingabe von Konstanten und Registern — Löschen von Registern — Addition, Subtraktion, Multiplikation, Division und Kehrwert von 8-Bit-Daten — Anhang.

Bestellnummer VV 577 DM 19,–

W. Schneider

Programmieren von Mikrocomputern 9

Einführung in die Anwendung des Betriebssystems CP/M
1983, 146 Seiten
Das Buch gibt eine grundlegende Einführung in das für 8-Bit-Mikrocomputer bevorzugte CP/M-Betriebssystem. Es werden alle die Kommandos besprochen, die der Anwender im Normalfall benötigt.
Aus dem Inhalt: Aufbau von Datenverarbeitungsanlagen — Programmiersprachen — Betriebssysteme — Grundlagen des CP/M-Betriebssystems — Starten des CP/M-Betriebssystems — CP/M-Dateinamen — Kopieren der CP/M-Betriebssystemdiskette — Das DIR-Kommando.

Bestellnummer VV 578 DM 14,90

N. Hoffmann

Anwendung von Mikrocomputern 1

Digitale Regelung mit Mikroprozessoren
1983, 169 Seiten
Das Buch ermöglicht, ein funktionsfähiges Mikroprozessor-Regelsystem zu entwerfen und zu realisieren. Vorausgesetzt wird die Kenntnis des Aufbaus und der Programmierung von Mikrocomputern.
Inhalt: Grundlagen der digitalen Regelung — Praktische Probleme: Verbindung von Prozeß und Rechner/Verschiedene Detailprobleme/Typen digitaler Regler/Systemplanung — Vollständige Dokumentation eines Regelsystems mit dem AIM-65 als Regler

Bestellnummer VV 435 DM 24,–

D. Herrmann

Anwendung von Mikrocomputern 2

30 BASIC-Programme
1983, 70 Seiten
Dieser Band stellt BASIC-Programme für die wichtigsten statistischen Tests bereit. Er liefert numerische Prozeduren für die Verteilungsfunktionen, so daß man völlig ohne Tabellenwerte auskommt. Berücksichtigt sind auch parametrische Tests, da viele Daten aus dem Bereich der Soziologie, Psychologie, Pädagogik usw. nicht hinreichend bekannten Verteilungen unterliegen.
Alle Programme verzichten auf spezielle Maschinenbefehle und sind damit auf alle BASIC-Computer übertragbar.

Bestellnummer VV 437 DM 11,40

H. Schumny

Mikroprozessoren

1983, 240 Seiten
Das Arbeitsbuch besteht aus zwei Teilen, die auch selbständig nutzbar sind. Teil 1 behandelt »Grundlagen und Basisoperationen« und stellt eine Einführung für Anfänger dar, ist aber auch als Repetitorium für Fortgeschrittene geeignet. Teil 2 hat die »Programmierung im Maschinencode« zum Inhalt. Dies ist also der eigentliche Arbeitsteil. Durch besondere grafische Gliederung wird dieser Teil dem Einsteiger das Arbeiten erleichtern. Fortgeschrittenen wird dadurch mit schnellem Zugriff das Wiederholen oder Vertiefen ermöglicht.

Bestellnummer VV 580 DM 24,–

99 Special I

1983, 298 Seiten
Das Buch führt vom spielerischen Beginn methodisch aufbauend den TI 99/4A-Anwender zu komplexer Programmierung. Programmbeispiele sind nach steigendem Schwierigkeitsgrad in die Kategorien Spiele, Mathematik, Datenorganisation, Grafik usw. unterteilt; Beispiel-Programmlistings runden den Inhalt ab. Für Leute, die bereits wissen, was Basic ist und die die ganze Palette von verfügbaren Programmiersprachen zum TI 99/4A erfahren wollen.

Bestellnummer TE 618 DM 24,75

6502-ANWENDUNGEN

1983, 288 Seiten, 205 Abbildungen
Das Eingabe-/Ausgabe-Buch für Ihren 6502-Mikroprozessor. Das Buch ist nach dem Konzept »Lernen durch Praxis« aufgebaut. Es stellt die notwendigen Hardwarekomponenten vor und die dafür meist genutzten Programme. Der Zweck eines jeden Programms, sein Flußdiagramm, die Schaltung und die im Programm verwendeten Programmiertechniken, werden eingehend beschrieben. Viele Anwendungsbeispiele helfen Ihnen, das Erlernte in die Praxis umzusetzen.

Bestellnummer SY 399 DM 19,–

Dr. J. Martin

Einführung in die Datenbanktechnik

1981, 369 Seiten
Dieses Buch eines weltberühmten Experten führt in didaktisch hervorragender Weise in die Datentechnik, die Datenbanktechnik sowie in das dazugehörige Umfeld ein. Es eignet sich wegen der systematischen Überblicksinformation sowohl für den fachlich schon tiefer Informierten als auch für diejenigen, die als verantwortliche Entscheidungsträger über diese Thematik Bescheid wissen müssen. Dem Autor ist es gelungen, dieses für unsere Zukunft so wichtige Gebiet fundiert darzustellen.

Bestellnummer CH 368 DM 29,–

Alle Preise inkl. MwSt. Der Versand erfolgt mit offener Rechnung zuzügl. Porto und Versand. Bitte benutzen Sie die Bestellkarte in diesem Heft!

```
100 REM*******************************        <181>
101 REM*          P-BASIC-V2.         *        <010>
102 REM*                              *        <073>
103 REM*   GESCHRIEBEN 1984  VON      *        <050>
104 REM*                              *        <075>
105 REM*          JAN KUSCH           *        <163>
106 REM*******************************        <187>
107 FOR X=0 TO 255:READ A:S=S+A               <229>
108 POKE 49152+X,A:NEXT                       <051>
109 IF S=26106 THEN 112                       <055>
110 PRINT"[DOWN5]?FEHLER[SPACE]IN[SPACE]DATAS
   [SPACE]:"                                  <211>
111 PRINT"PRUEFSUMME[SPACE]:"S"[SPACE2](26106)"
   :END                                       <048>
112 POKE 49152+X,ASC ("!")                    <182>
113 SYS 49152:NEW                             <240>
114 DATA 169,17,141,8,3,169,192,141,9,3,169,49,
   160,192,76,30,171                          <165>
115 DATA 32,115,0,205,0,193,240,3,76,231,167,32,
   115,0,201,147,240                          <161>
116 DATA 64,201,148,240,88,201,162,240,3,76,8,
   175,76,180,192,147,144                     <101>
117 DATA 18,32,208,45,194,65,83,73,67,45,214,50,
   46,32,32,13,195                            <112>
118 DATA 45,54,52,32,38,32,214,195,45,49,53,52,
   49,13,195,79,80                            <074>
119 DATA 89,82,73,71,72,84,32,40,67,41,9,14,8,0,
   169,0,133                                  <068>
120 DATA 157,32,115,0,32,212,225,169,0,166,43,
   164,44,133,10,32,213                       <225>
121 DATA 255,134,45,132,46,76,180,192,169,0,133,
   157,32,115,0,32,212                        <037>
122 DATA 225,162,0,189,218,192,157,64,3,232,201,
   255,208,245,169,38,133                     <207>
123 DATA 43,169,3,133,44,169,64,141,36,3,141,38,
   3,169,3,141,37                             <058>
124 DATA 3,141,39,3,32,68,229,32,89,225,169,1,
   133,43,169,8,133                           <068>
125 DATA 44,169,87,141,36,3,169,241,141,37,3,
   141,39,3,169,202,141                       <210>
126 DATA 38,3,169,237,141,40,3,169,0,133,198,76,
   49,168,169,1,133                           <178>
127 DATA 43,169,8,133,44,169,87,141,36,3,169,
   241,141,37,3,141,39                        <173>
128 DATA 3,169,202,141,38,3,169,225,141,40,3,32,
   89,166,76,174,167                          <222>
129 DATA 255                                  <160>
```

```
100 SYS 9*4096 : .OPT OO
101 *=49152
102 ;
103 ;
104 ;
105 ;#################################
106 ;* * *  P - BASIC - V2.    * * *
107 ;#################################
108 ;
109 ; *****************************
110 ;*GESCHRIEBEN1984V ON*
111 ;**
112 ;*JANKUSCH(J-KSOFT)*
113 ;**
114 ; *****************************
115 ;
116 ;
117 ;#################################
118 ;####### VEK TO RSETZUNG ########
119 ;#################################
120 ;
121 LDA #<ERKENNUNG    : STA $0308
122 LDA #>ERKENNUNG    : STA $0309
123 ;
124 LDA #<DATA S         : LDY #>DATA S
125 JMP $AB1E
126 ;
127 ;
128 ;#################################
129 ;######### ERKENNUNG ##########
130 ;#################################
131 ERKENNUNG JSR $0073 : CMP BEFEHL
132 BEQ ENTSCHEIDUNG
133 JMP $A7E7
134 ;
135 ENTSCHEIDUNG JSR $0073
136 CMP #147    : BEQ LOAD
137 CMP #148    : BEQ SAVE
138 CMP #162    : BEQ NEW 1
139 JMP $AF08
140 ;
141 NEW 1          JMP NEW
142 ;
143 ;
144 ;#################################
145 ;############ DATA S ############
146 ;#################################
147 DATA S .ASC "[CLEAR,BLACK,RVSON,SPACE]
   P-BASIC-V2.[SPACE2]"
148 .BYTE 13 : .ASC "C-64[SPACE]&[SPACE]VC-1541"
149 .BYTE 13 : .ASC "COPYRIGHT[SPACE](C)"
150 .BYTE 9,14,8,0
151 ;
152 ;#################################
153 ;######## BEFEHL .LOAD ########
154 ;#################################
155 LOAD LDA #0 : STA $9D
156 JSR $0073   : JSR $E1D4
157 LDA #00     : LDX $2B  : LDY $2C
158 STA $0A     : JSR $FFD5
159 STX $2D     : STY $2E
160 JMP NEW
161 ;
162 ;#################################
163 ;####### BEFEHL .SAVE #########
164 ;#################################
165 SAVE LDA #00
166 STA $9D      : JSR $0073 : JSR $E1D4
167 ;
168 LDX #00
169 VERSCHIEBUNG  LDA ROUTINE,X
170 STA $0340,X : INX
171 CMP #$FF     : BNE VERSCHIEBUNG
172 ;
173 LDA #<806   : STA $2B
174 LDA #>806   : STA $2C
175 LDA #<832   : STA 804    : STA 806
176 LDA #>832   : STA 805    : STA 807
177 ;
178 JSR $E544   : JSR $E159
179 ;
180 NEW LDA #1  : STA $2B
181 LDA #8      : STA $2C
182 LDA #$57    : STA 804
183 LDA #$F1    : STA 805    : STA 807
184 LDA #$CA    : STA 806
185 LDA #237    : STA 808
186 LDA #0      : STA 198    : JMP $A831
187 ;
188 ;
189 ;******* STEUERROUTIENE *******
190 ;
191 ROUTINE LDA #1 : STA $2B
192 LDA #8         : STA $2C
193 LDA #$57       : STA 804
194 LDA #$F1       : STA 805 : STA 807
195 LDA #$CA       : STA 806
196 LDA #225       : STA 808
197 ;
198 JSR $A659      : JMP $A7AE
199 ;
200 .BYTE $FF
201 ;
202 BEFEHL .BYTE "!"
203 ;
```

Toolkit für Programmierer

Nicht jeder kann sich professionelle Toolkits wie zum Beispiel Ex-Basic Level II leisten. Dieses Programm hat deshalb einige der wichtigsten Funktionen, die das Programmieren sehr erleichtern.

Dieses Programm für den Commodore 64 bietet 8 leistungsfähige Befehle und 2 zusätzliche Funktionen, die bei der Programmerstellung und beim Austesten von Programmen besonders nützlich sind. Hier die Erklärung zu den einzelnen Befehlen:

!AUTO X,Y

Dieser Befehl erspart Ihnen das Eintippen der Zeilennummern. Mit X geben Sie die Anfangszeilennummer ein. Mit Y die Schrittweite, um welcher die Zeilennummern erhöht werden sollen. X kann eine positive Integerzahl sein. Y kann im Bereich von 0-255 liegen. Wenn Sie den AUTO-Befehl eingetippt haben, erscheint nach Drücken der C= Taste zusammen mit der Taste F7 die erste Zeilennummer. Schreiben Sie nun Befehle hinter die Zeilennummer und schließen Sie mit RETURN ab. Wenn Sie nun wieder die Taste C= und F7 drücken, erscheint eine um Y erhöhte Zeilennummer. Abgeschaltet wir die AUTO-Funktion mit der C= Taste zusammen mit F1. Außerdem wird diese Funktion abgeschaltet, wenn Sie RUN/STOP und RESTORE drücken.

!DUMP

Damit können Sie sich sämtliche Variablen mit ihrem aktuellen Inhalt ansehen. Indizierte Variablen werden nicht mit ausgegeben. Das Auflisten der Variableninhalte kann mit der CTRL-Taste verlangsamt, oder mit der STOP-Taste abgebrochen werden.

!FIND

Hierbei muß unterschieden werden zwischen dem Suchen in Zeichenketten und dem Suchen von Befehlen und Variablen. Zum Beispiel: »!FIND "TEST"«; »!FIND GOTO1000« oder »!FIND A$«. Es werden jeweils die kompletten Zeilen gelistet in denen der Suchbegriff vorkommt. Dieser Vorgang kann wie bei dem DUMP -Befehl mit der CTRL - Taste verlangsamt oder mit STOP ganz abgebrochen werden.

!KILL

Löscht statt einzelne Zeilen ganze Zeilenblöcke.
Beispiel:

!KILL 100-200	—	löscht die Zeilen 100 bis einschließlich 200.
!KILL -300	—	löscht die Zeilen vom Programmanfang bis einschließlich 300.
!KILL 1800-	—	löscht die Zeilen ab 1800 bis zum Programmende.

Bei diesem Befehl muß folgendes beachtet werden: Die erste Zeilennummer muß kleiner als die Endzeilennummer sein. Weiter ist notwendig, daß die Anfangszeilennummer im Programm existiert. Andernfalls kommt die Fehlermeldung 'SYNTAX ERROR'. Bei umfangreicheren Zeilenblöcken dauert die Ausführung dieses Befehls eine Weile. Mit der Taste 'F7' kann das Löschen von Zeilenblöcken abgebrochen weden.

!LOAD

Dieser Befehl hängt an einem im Speicher befindlichen Programm ein weiteres an. Die Syntax ist wie bei dem LOAD-Befehl. Es ist vorteilhaft, wenn das nachzuladende Programm höhere Zeilennummern hat, aber nicht unbedingt notwendig. Sie haben ja mit dem RENUM-Befehl die Möglichkeit das Programm neu zu numerieren. Nur bei gleichen Zeilennummern kann es zu Fehlern beim Umnumerieren der Sprungbefehle kommen.

!RENUM

Hiermit numerieren sie Ihr Programm mit einer von Ihnen angegebenen Schrittweite neu durch. Sie haben bei diesem Befehl mehrere Möglichkeiten.
!RENUM — ohne weitere Parameter numeriert das komplette Programm in Zehnerschritten durch. Die erste Zeilennummer ist 10.
!RENUM X,Y — Hierbei ist X die erste neue Zeile und Y die Schrittweite.
!RENUM X,Y,Z — Damit numerieren Sie nicht das komplette Programm neu, sondern ab der Zeile die in Z angegeben ist. Z muß dabei kleiner als X sein. Als Parameter werden jeweils positive Integerzahlen erwartet. Selbstverständlich werden sämtliche Sprungbefehle (GOTO, GOSUB, THEN, ON X GOTO (GOSUB), RUN) mit umnumeriert.

!TRACE

Damit können Sie ein Programm schrittweise abarbeiten. Nachdem Sie !TRACE eingegeben und ein Programm mit RUN gestartet haben, wird in den oberen beiden Zeilen des Bildschirms die Zeile gelistet, die gerade bearbeitet wird. Damit Sie sehen, welcher Befehl in dieser Zeile gerade ausgeführt wird, wird dieser revers dargestellt. Zur Steuerung der TRACE-Funktion stehen Ihnen die Tasten F1 und F7 zur Verfügung. Mit F1 wird ein schneller Trace ausgeführt. Mit F7 wird immer nur der nächste Befehl bearbeitet. Verlassen können Sie die TRACE-Funktion, wenn Sie beim schnellen Trace die STOP-Taste drücken.

!OFF

schaltet die TRACE-Funktion aus.

Sämtliche Befehle können auch innerhalb eines Programms benutzt werden. Sinnvoll ist dies aber wohl nur bei !DUMP, !TRACE und !OFF. Zu den zwei zusätzlichen Funktionen:
1. Trifft der Computer während des Programmlaufs auf einen Fehler, so wird automatisch die fehlerhafte Zeile gelistet. Der Cursor steht direkt unter der fehlerhaften Zeile.
2. Beim Listen eines Programms können Sie den Listvorgang mit der Taste F7 anhalten. Sie können sich Ihr Listing nun in Ruhe betrachten. Soll der Listvorgang fortgeführt werden, so drücken Sie die SPACE-Taste. Ein mit der Taste F7 angehaltenes Listing kann mit der Taste F1 unterbrochen werden. Die STOP-Taste steht hierfür nicht zur Verfügung. Bei der Taste F1 wird BREAK und READY (wie es bei der STOP-Taste der Fall ist) nicht mit ausgegeben. Sie haben so mehr Platz für Ihr Listing. Zu dem Programm selbst. Es steht in dem Bereich ab Speicherstelle 49152. Dieser Bereich wird vom Basic nicht benutzt. Aktiviert wird das Programm mit SYS 51100. Wollen Sie ein anderes Programm in diesem Bereich nutzen, so müssen Sie das Toolkit abschalten. Dies erreichen Sie mit einem System-Reset (SYS 64738).

Nach erfolgreichem Eintippen sollten Sie das Programm (Bild 1) mit einem Maschinensprachenmonitor speichern. Die Ladezeit verringert sich dadurch erheblich. Besitzen Sie keinen solchen Monitor, so benutzen Sie das kurze Programm in Bild 2.

(Herbert Kunz / gk)

```
10 REM  ******************************     <091>
20 REM                                      <163>
30 REM     TOOL-KIT FUER  C= 64            <028>
40 REM                                      <183>
50 REM          HERBERT KUNZ               <169>
60 REM          FIMBERGSTR. 28             <088>
70 REM                                      <213>
80 REM          3163 SEHNDE 1              <148>
90 REM                                      <233>
98 REM  ******************************     <179>
99 REM                                      <242>
100 REM   *****   BLOCK 1   *****          <051>
101 DATA 32,32,32,32,32,32,32,32,32,32     <102>
102 DATA 32,42,42,42,32,84,79,79,76,45     <147>
103 DATA 75,73,84,32,42,42,42,13,32,32     <125>
104 DATA 32,32,32,32,32,32,32,32,32,40     <104>
105 DATA 67,41,32,72,46,32,75,85,78,90     <152>
106 DATA 32,49,57,56,52,32,32,84,82,65     <148>
107 DATA 67,69,35,68,85,77,80,35,35,79     <175>
108 DATA 70,70,35,35,35,70,73,78,68,35     <151>
109 DATA 35,82,69,78,85,77,35,65,85,164    <226>
110 DATA 35,35,35,75,73,76,76,35,0,193     <154>
111 DATA 58,194,11,193,146,195,123,197     <161>
112 DATA 250,194,109,196,32,68,229,162     <159>
113 DATA 255,232,189,16,192,32,210,255     <148>
114 DATA 224,55,208,245,169,155,141,8      <105>
115 DATA 3,169,192,141,9,3,96,32,115       <050>
116 DATA 0,201,33,240,6,32,121,0,76,231    <164>
117 DATA 167,169,73,133,80,169,192,133     <166>
118 DATA 81,160,0,140,7,192,76,98,200      <097>
119 DATA 209,80,240,17,170,152,24,105      <091>
120 DATA 6,201,42,240,20,238,7,192,168     <145>
121 DATA 138,76,184,192,200,177,80,201     <158>
122 DATA 35,240,10,32,115,0,209,80,240     <129>
123 DATA 242,76,8,175,173,7,192,10,170     <159>
124 DATA 189,114,192,141,241,192,232       <059>
125 DATA 189,114,192,141,242,192,32,146    <210>
126 DATA 195,76,132,195,234,234,234,234    <218>
127 DATA 234,234,234,234,234,234,169       <064>
128 DATA 22,141,8,3,169,193,141,9,3,96     <162>
129 DATA 169,155,141,8,3,169,192,141       <070>
130 DATA 9,3,96,165,157,240,3,76,155       <074>
131 DATA 192,165,122,141,0,192,165,123     <155>
132 DATA 141,1                             <250>
133 S=0:FOR I= 49168 TO 49445 :READ D      <047>
134 POKE I,D:S=S+D:NEXT                    <018>
135 IF S<>29085 THEN PRINT"FEHLER[SPACE]IN
    [SPACE]BL.[SPACE]1":STOP               <089>
200 REM   *****   BLOCK 2   *****          <152>
201 DATA 192,165,211,141,2,192,165,214     <227>
202 DATA 141,3,192,32,135,234,165,203      <174>
203 DATA 201,3,208,20,32,135,234,165       <119>
204 DATA 203,201,64,208,247,169,0,141      <178>
205 DATA 232,7,32,121,193,76,95,193,201    <031>
206 DATA 4,208,5,169,1,141,232,7,173       <132>
207 DATA 232,7,240,213,32,121,193,173      <177>
208 DATA 0,192,133,122,173,1,192,133       <126>
209 DATA 123,173,2,192,133,211,173,3       <128>
210 DATA 192,133,214,32,108,229,76,155     <246>
211 DATA 192,169,0,133,15,165,57,133       <146>
212 DATA 20,133.99,165,58,133,21,133       <144>
213 DATA 98,32,[illegible],166,169,80,141,4,192     <003>
214 DATA 32,102,229,169,32,32,210,255      <187>
215 DATA 206,4,192,173,4,192,208,243       <148>
216 DATA 32,102,229,32,209,189,160,3       <144>
217 DATA 169,32,132,73,32,210,255,164      <196>
218 DATA 73,177,95,201,34,208,6,165,15     <000>
219 DATA 73,255,133,15,200,132,73,177      <198>
220 DATA 95,208,1,96,201,255,240,227       <155>

221 DATA 36,15,48,223,165,73,56,233,1      <204>
222 DATA 24,101,95,141,5,192,176,8,165     <000>
223 DATA 96,141,6,192,76,235,193,165       <173>
224 DATA 96,24,105,1,141,6,192,173,5       <151>
225 DATA 192,205,0,192,208,13,173,6,192    <046>
226 DATA 205,1,192,208,5,169,18,32,210     <249>
227 DATA 255,164,73,177,95,16,38,56,233    <073>
228 DATA 127,170,160,255,202,240,8,200     <244>
229 DATA 185,158,160,16,250,48,245,200     <006>
230 DATA 185,158,160,48,5,32,210,255       <165>
231 DATA 208,245,41,127,32,210,255,169     <006>
232 DATA 146,76,173,193,165,15,240,5       <172>
233 DATA 177,95,76,36,194,177,95,76,173    <098>
234 DATA 193,165,45                        <103>
235 S=0:FOR I= 49446 TO 49723 :READ D      <147>
236 POKE I,D:S=S+D:NEXT                    <120>
237 IF S<>35613 THEN PRINT"FEHLER[SPACE]IN
    [SPACE]BL.[SPACE]2":STOP               <186>
300 REM   *****   BLOCK 3   *****          <254>
301 DATA 164,46,133,20,132,21,196,48       <232>
302 DATA 208,2,197,47,176,24,105,2,144     <079>
303 DATA 1,200,133,34,132,35,32,128,194    <112>
304 DATA 32,180,194,138,16,7,32,189,194    <143>
305 DATA 76,111,194,96,152,48,6,32,205     <088>
306 DATA 194,76,111,194,32,214,194,169     <096>
307 DATA 13,32,145,200,165,20,164,21       <220>
308 DATA 24,105,7,144,193,200,176,190      <031>
309 DATA 160,0,177,20,170,41,127,32,210    <112>
310 DATA 255,200,177,20,168,41,127,240     <081>
311 DATA 3,32,210,255,138,16,17,152,48     <081>
312 DATA 10,169,64,32,210,255,104,104      <025>
313 DATA 76,111,194,169,37,208,78,152      <058>
314 DATA 16,4,169,36,208,71,96,32,210      <043>
315 DATA 255,169,32,32,210,255,169,61      <047>
316 DATA 208,58,160,0,177,34,170,200       <242>
317 DATA 177,34,168,138,32,149,179,76      <071>
318 DATA 208,194,32,166,187,32,221,189     <108>
319 DATA 76,30,171,32,245,194,160,2,177    <146>
320 DATA 34,133,37,136,177,34,133,36       <000>
321 DATA 136,177,34,133,38,240,10,177      <049>
322 DATA 36,32,210,255,200,196,38,208      <046>
323 DATA 246,169,34,76,210,255,32,115      <054>
324 DATA 0,32,235,183,142,235,7,165,20     <089>
325 DATA 166,21,141,233,7,142,234,7,120    <136>
326 DATA 169,26,162,195,141,20,3,142       <000>
327 DATA 21,3,88,96,173,138,2,141,10       <000>
328 DATA 192,169,64,141,138,2,32,135       <008>
329 DATA 234,173,141,2,201,2,240,9,169     <093>
330 DATA 0,234,141,138,2,76,49,234,165     <106>
331 DATA 203,201,3,240,7,201,4,240,38      <031>
332 DATA 76,47,195,32,122,195,173,234      <072>
333 DATA 7,174,233,7,32,205,189,169        <229>
334 S=0:FOR I= 49724 TO 50001 :READ D      <227>
335 POKE I,D:S=S+D:NEXT                    <220>
336 IF S<>32656 THEN PRINT"FEHLER[SPACE]IN
    [SPACE]BL.[SPACE]3":STOP               <035>
400 REM   *****   BLOCK 4   *****          <099>
401 DATA 29,32,210,255,173,233,7,24,109    <222>
402 DATA 235,7,141,233,7,144,3,238,234     <174>
403 DATA 7,76,47,195,32,122,195,120,169    <241>
404 DATA 49,162,234,141,20,3,142,21,3      <111>
405 DATA 88,76,47,195,32,135,234,165       <103>
406 DATA 203,201,64,208,247,96,169,2       <088>
407 DATA 56,237,7,192,144,3,32,115,0       <077>
408 DATA 76,174,167,32,115,0,240,125       <081>
409 DATA 201,34,240,17,162,0,142,3,192     <164>
410 DATA 157,0,207,32,115,0,240,35,232     <165>
411 DATA 76,160,195,160,1,140,3,192,162    <231>
412 DATA 255,177,122,201,34,240,8,200      <126>
413 DATA 232,157,0,207,76,179,195,32       <100>
414 DATA 115,0,201,34,208,249,76,163       <087>
415 DATA 195,232,169,0,157,0,207,160       <090>

416 DATA 0,177,43,141,0,192,200,177,43     <180>
417 DATA 141,1,192,165,43,133,95,165       <096>
418 DATA 44,133,96,76,96,196,173,3,192     <218>
419 DATA 141,2,192,160,0,177,95,141,0      <133>
420 DATA 192,200,177,95,141,1,192,173      <149>
421 DATA 0,192,133,95,173,1,192,133,96     <200>
422 DATA 32,208,248,160,0,177,95,208       <105>
423 DATA 6,200,177,95,208,1,96,160,3       <100>
424 DATA 162,0,200,177,95,240,203,201      <134>
425 DATA 34,208,8,169,1,77,2,192,141       <105>
426 DATA 2,192,173,2,192,208,232,177       <105>
427 DATA 95,221,0,207,208,225,232,189      <155>
428 DATA 0,207,240,3,76,25,196,160,0       <093>
429 DATA 132,15,200,200,177,95,133,20      <139>
430 DATA 133,99,200,177,95,133,21,133      <159>
431 DATA 98,169,13,32,210,255,32,19,166    <008>
432 DATA 32,164,193,76,233,195,173,3       <119>
433 DATA 192,141,2,192,76,6,196,234        <069>
434 S=0:FOR I= 50002 TO 50281 :READ D      <062>
435 POKE I,D:S=S+D:NEXT                    <064>
436 IF S<>34230 THEN PRINT"FEHLER[SPACE]IN
    [SPACE]BL.[SPACE]4":STOP               <126>
500 REM   *****   BLOCK 5   *****          <200>
501 DATA 76,8,175,32,115,0,240,248,201     <014>
502 DATA 171,208,19,160,2,177,43,141       <176>
503 DATA 0,192,200,177,43,141,1,192,32     <007>
504 DATA 115,0,76,159,196,32,107,169       <192>
505 DATA 166,20,142,0,192,166,21,142       <170>
506 DATA 1,192,201,171,208,208,32,115      <219>
507 DATA 0,240,16,32,107,169,165,20,141    <061>
508 DATA 2,192,165,21,141,3,192,76,183     <027>
```

```
509 DATA 196,169,254,141,2,192,141,3          <189>
510 DATA 192,173,3,192,56,237,1,192,144       <085>
511 DATA 170,208,11,173,2,192,56,237          <187>
512 DATA 0,192,144,159,240,157,173,0          <187>
513 DATA 192,133,20,173,1,192,133,21          <177>
514 DATA 32,19,166,144,142,169,233,141        <037>
515 DATA 2,3,169,196,141,3,3,76,10,197        <036>
516 DATA 173,3,192,56,237,1,192,144,11        <034>
517 DATA 208,22,173,2,192,56,237,0,192        <036>
518 DATA 176,13,169,131,162,164,141,2         <242>
519 DATA 3,142,3,3,76,134,227,173,0,192       <079>
520 DATA 133,20,173,1,192,133,21,32,19        <023>
521 DATA 166,160,0,177,95,133,247,200         <249>
522 DATA 177,95,133,248,177,247,208,8         <019>
523 DATA 169,255,141,1,192,76,59,197          <223>
524 DATA 160,2,177,247,141,0,192,200          <190>
525 DATA 177,247,141,1,192,32,135,234         <252>
526 DATA 165,203,201,3,240,185,160,2          <187>
527 DATA 177,95,133,99,200,177,95,133         <020>
528 DATA 98,162,144,56,32,73,188,32,223       <111>
529 DATA 189,32,135,180,32,166,182,170        <056>
530 DATA 160,0,232,202,240,10,177,34          <186>
531 DATA 153,0,2,200,201,13,208,243,169       <079>
532 DATA 0,153,0,2,162,255,160,1,76,134       <082>
533 DATA 164                                  <053>
534 S=0:FOR I= 50282 TO 50554 :READ D         <176>
535 POKE I,D:S=S+D:NEXT                       <165>
536 IF S<>33081 THEN PRINT"FEHLER[SPACE]IN
   [SPACE]BL.[SPACE]5":STOP                   <231>
600 REM   *****   BLOCK 6   *****             <046>
601 DATA 32,115,0,240,64,32,107,169,165       <163>
602 DATA 20,166,21,133,61,134,62,32,253       <158>
603 DATA 174,32,107,169,165,20,166,21         <075>
604 DATA 133,63,134,64,160,0,169,44,209       <173>
605 DATA 122,208,17,32,253,174,32,107         <069>
606 DATA 169,165,20,166,21,133,65,134         <079>
607 DATA 66,76,207,197,160,2,177,43,133       <189>
608 DATA 65,200,177,43,133,66,76,220          <032>
609 DATA 197,169,10,133,61,133,63,169         <090>
610 DATA 0,133,62,133,64,76,178,197,165       <193>
611 DATA 61,197,65,165,62,229,66,176          <058>
612 DATA 3,76,113,199,32,117,199,160          <045>
613 DATA 1,177,67,240,56,169,255,133          <047>
614 DATA 20,133,21,32,50,199,32,142,166       <173>
615 DATA 76,41,198,160,0,177,67,170,200       <189>
616 DATA 177,67,240,31,72,200,165,61          <037>
617 DATA 145,67,200,165,62,145,67,134         <094>
618 DATA 67,104,133,68,165,61,24,101          <037>
619 DATA 63,133,61,165,62,101,64,133          <034>
620 DATA 62,144,215,32,96,166,76,134          <052>
621 DATA 227,230,122,208,2,230,123,160        <121>
622 DATA 0,177,122,208,19,160,2,177,122       <184>
623 DATA 240,191,165,122,24,105,5,133         <082>
624 DATA 122,144,235,230,123,176,231          <036>
625 DATA 201,34,208,11,32,115,0,201,0         <059>
626 DATA 240,226,201,34,208,245,201,137       <184>
627 DATA 240,23,201,141,240,19,201,167        <129>
628 DATA 240,15,201,138,240,11,201,203        <120>
629 DATA 208,190,32,115,0,201,164,208         <086>
630 DATA 191,32,115,0,176,186,132,20          <043>
631 DATA 132,21,233,47,144,51,170,165         <095>
632 DATA 21,133,34,201,25,176,237,165         <098>
633 DATA 20,10,38,34,10,38                    <059>
634 S=0:FOR I= 50555 TO 50824 :READ D         <023>
635 POKE I,D:S=S+D:NEXT                       <009>
636 IF S<>32153 THEN PRINT"FEHLER[SPACE]IN
   [SPACE]BL.[SPACE]6":STOP                   <075>
700 REM   *****   BLOCK 7   *****             <147>
701 DATA 34,101,20,133,20,165,34,101          <095>
702 DATA 21,133,21,6,20,38,21,138,101         <148>
703 DATA 20,133,20,144,2,230,21,200,177       <243>
704 DATA 122,201,32,240,249,201,58,144        <212>
705 DATA 201,165,20,197,65,165,21,229         <179>
706 DATA 66,144,62,132,206,32,50,199          <131>
707 DATA 56,162,144,32,73,188,32,221          <131>
708 DATA 189,160,255,200,185,1,1,208          <126>
709 DATA 250,165,122,133,95,165,123,133       <020>
710 DATA 96,56,152,229,206,48,28,240          <146>
711 DATA 54,133,206,165,45,133,90,24          <131>
712 DATA 101,206,133,88,165,46,133,91         <185>
713 DATA 105,0,133,89,32,184,163,240          <130>
714 DATA 30,144,48,73,255,168,200,162         <187>
715 DATA 0,165,95,197,45,165,96,229,46        <006>
716 DATA 176,12,177,95,129,95,230,95          <160>
717 DATA 208,238,230,96,208,234,160,0         <187>
718 DATA 185,1,1,240,5,145,122,200,208        <217>
719 DATA 246,32,117,199,32,115,0,144          <138>
720 DATA 251,160,0,201,44,208,2,169,137       <022>
721 DATA 170,76,45,198,165,61,166,62          <162>
722 DATA 133,99,134,98,165,67,166,68          <177>
723 DATA 133,34,134,35,160,2,177,34,197       <038>
724 DATA 20,200,177,34,229,21,176,37          <143>
725 DATA 160,0,177,34,170,200,177,34          <139>
726 DATA 134,34,133,35,177,34,240,21          <140>
727 DATA 165,99,101,63,133,99,165,98          <175>
728 DATA 101,64,133,98,176,4,201,250          <145>
729 DATA 144,209,76,72,178,96,32,51,165       <062>
730 DATA 24,165,34,105,2,133,45,165,35        <244>
731 DATA 105,0,133,46,165,65,166,66,133       <046>
732 DATA 20,134,21,32,19,166,165,95,133       <043>
733 DATA 67,134,68,96                         <200>
734 S=0:FOR I= 50825 TO 51094 :READ D         <123>
735 POKE I,D:S=S+D:NEXT                       <109>
736 IF S<>31170 THEN PRINT"FEHLER[SPACE]IN
   [SPACE]BL.[SPACE]7":STOP                   <174>
800 REM   *****   BLOCK 8   *****             <217>
801 DATA 169,179,141,0,3,169,199,141          <238>
802 DATA 1,3,169,37,141,6,3,169,200,141       <105>
803 DATA 7,3,76,128,192,138,16,3,76,116       <126>
804 DATA 164,234,138,10,170,189,38,163        <078>
805 DATA 133,34,189,39,163,133,35,32          <233>
806 DATA 204,255,169,0,133,19,32,215          <224>
807 DATA 170,32,69,171,160,0,177,34,72        <072>
808 DATA 41,127,32,71,171,200,104,16          <213>
809 DATA 244,32,122,166,169,105,160,163       <123>
810 DATA 32,30,171,164,58,200,240,3,32        <057>
811 DATA 194,189,169,118,160,163,32,30        <088>
812 DATA 171,165,157,240,8,169,128,32         <038>
813 DATA 144,255,76,128,164,165,57,166        <100>
814 DATA 58,133,20,133,99,134,21,134          <234>
815 DATA 98,32,19,166,169,0,133,15,32         <036>
816 DATA 164,193,169,13,32,210,255,76         <039>
817 DATA 255,199,141,15,192,140,14,192        <088>
818 DATA 165,145,141,9,192,32,135,234         <036>
819 DATA 165,203,201,3,240,14,172,14          <222>
820 DATA 192,173,9,192,133,145,173,15         <044>
821 DATA 192,76,26,167,32,135,234,165         <049>
822 DATA 203,201,4,240,10,201,60,208          <214>
823 DATA 243,169,0,133,198,240,223,169        <096>
824 DATA 13,32,210,255,76,128,164,32          <242>
825 DATA 115,0,201,147,240,3,76,184,192       <132>
826 DATA 164,46,165,45,56,233,2,176,1        <046>
827 DATA 136,140,1,192,141,0,192,169          <243>
828 DATA 0,133,10,32,115,0,32,212,225        <009>
829 DATA 165,10,174,0,192,172,1,192,76        <092>
830 DATA 117,225,32,210,255,76,208,248        <099>
831 S=0:FOR I= 51100 TO 51350 :READ D         <203>
832 POKE I,D:S=S+D:NEXT                       <207>
833 IF S<>29856 THEN PRINT"FEHLER[SPACE]IN
   [SPACE]BL.8":STOP                          <035>
840 :                                         <133>
845 :                                         <138>
850 SYS 51100:REM TOOL-KIT AKTIVIEREN         <248>
```

Bild 1. Der Basic-Lader erzeugt das Toolkit. Speichern Sie dieses Programm sicherheitshalber vor dem RUN ab. Nach dem Start benutzen Sie das Programm in Bild 2, um den Maschinencode zu speichern.

```
5 REMTOOL-KIT
10 POKE183,8:REM LAENGE DES FILENAMENS
20 POKE185,0:REM SEKUNDAERADRESSE
30 POKE186,1:REM GERAETEADRESSE (8 FUER FLOPPY)
40 POKE187,PEEK(43)+5:POKE188,PEEK(44):REM ZEIGER AUF FILENAMEN
50 POKE193,16:POKE194,192:REM HI- UND LOW BYTE STARTADRESSE
60 POKE174,151:POKE175,200:REM HI- UND LOW BYTE ENDADRESSE +1
70 SYS62954:REM ZUR SAVE - ROUTINE
80 REM LADEN MIT LOAD"TOOL-KIT",1,1 (,8,1 FUER FLOPPY)
90 REM NACH DEM LADEN NEW EINGEBEN
95 REM DANN MIT SYS 51100 TOOL-KIT AKTIVIEREN
```

Bild 2. Nach erfolgreichem Abtippen und Start des Basic-Laders können Sie das Toolkit mit diesem Programm abspeichern.

Eigene Basic-Befehle auf dem C 64

Bereits implementiert sind Befehle, wie Text und hochauflösende Grafik zu kombinieren oder Hintergrundfarbe ändern. Sie können bis zu 51 neue Befehle definieren.

Das Programm ermöglicht es, eigene Basic-Befehle zu definieren. Als Beispiel sind sechs Befehle definiert worden, die den hochauflösenden Grafikmodus betreffen. Alle Anweisungen, die benötigt werden, sind in REM-Zeilen gespeichert.

(Förtsch / rg)

```
50000 REM *************************           <230>
50010 REM *                       *           <000>
50020 REM *   BEFEHLSERWEITERUNG  *           <084>
50040 REM *   ------------------  *           <072>
50050 REM *                       *           <040>
50060 REM *       1984 BY         *           <163>
50070 REM *                       *           <060>
50080 REM *   G. FOERTSCH         *           <025>
50090 REM *   DUEPPEL-STR. 84     *           <031>
50100 REM *   5100 AACHEN         *           <192>
50110 REM *                       *           <100>
50120 REM *   TEL.: 0241/507569   *           <241>
50130 REM *                       *           <120>
50140 REM *************************           <114>
50150 REM                                     <056>
50200 REM ZUNAECHST FOLGT EINE ERLAEU-
      TERUNG DER AB ZEILE 60860               <088>
50210 REM ALS BEISPIELE AUFGEFUEHRTEN
      BEFEHLSERWEITERUNGEN.                   <187>
50220 REM
      BEFEHL (1): GRAFIC                      <141>
50230 REM DER HOCHAUFLOESENDE GRAFIK-
      MODUS WIRD EINGESCHALTET. DER           <025>
50240 REM GRAFIKSPEICHER BEGINNT BEI
      $2000 (DEZ8192).                        <141>
50250 REM
      BEFEHL (2): GOFF                        <034>
50260 REM DER GRAFIKMODUS WIRD AUSGE-
      SCHALTET.                               <028>
50270 REM
      BEFEHL (3): GCLEAR                      <195>
50280 REM DER GRAFIKSPEICHER WIRD GE-
      LOESCHT. (FALLS SIE DIESEN              <255>
50290 REM GANZEN REM-RAMSCH ABGETIPPT
      HABEN, IST DIESER BEFEHL MIT            <189>
50300 REM AEUSSERSTER VORSICHT ZU GE-
      BRAUCHEN, DA DAS PROGRAMM WE-           <254>
50310 REM GEN SEINER LAENGE ZERSTOERT
      WIRD!!!)                                <185>
50320 REM
      BEFEHL (4): SCOLOR(PF,HF)               <187>
50330 REM IM VIDEORAM WIRD DIE FARBE GE-
      SETZT MIT:                              <185>
50340 REM  PF...PUNKTFARBE (0-15)
       HF...HINTERGRUNDFARBE (0-15)           <243>
50350 REM
      BEFEHL (5): GTEXT                       <243>
50360 REM DIESER BEFEHL GESTATTET ES IH-
      NEN, TEXT UND GRAFIK ZU                 <020>
50370 REM MISCHEN. GTEXT SCHALTET DIESEN
      MODUS EIN. MIT GTEXT(X1,X2)             <126>
50380 REM KOENNEN SIE ZWEI PARAMETER
      UEBERGEBEN, DIE DIE LAGE DES            <135>
50390 REM GRAFIKFENSTERS AUF DEM BILD-
      SCHIRM FESTLEGEN. FUER X1               <141>
50400 REM BZW. X2 DUERFEN NUR WERTE
      ZWISCHEN 0 UND 25 GEWAEHLT              <157>
50410 REM WERDEN. BEISPIEL: GTEXT(3,
   20).          IN DEN ERSTEN 3 BILDSCHIRM-
                       <003>
50420 REM ZEILEN UND AB ZEILE 20 KANN
      TEXT UNTERGEBRACHT WERDEN,              <127>
50430 REM DAZWISCHEN LIEGT DAS GRAFIK-
      FENSTER. (SOLLTEN SICH DER              <255>
50440 REM TEXT UND DIE KANTEN DES FEN-
      STERS UEBERLAPPEN, MUESSTEN             <097>
50450 REM SIE DAS DATUM IN ZEILE 61441

      ETWAS VARIIEREN.)                       <042>
50460 REM
      BEFEHL (6): TEXTOFF                     <247>
50470 REM OBEN BESCHRIEBENER MODUS WIRD
      AUSGESCHALTET.                          <008>
50480 REM                                     <132>
50490 REM WENN SIE LUST VERSPUEREN,
    DIE-            SE BEFEHLE AUSZUPROBIEREN,
                       <022>
50500 REM STARTEN SIE DAS PROGRAMM MIT
      RUN 60800 UND INITIALISIEREN            <180>
50510 REM SIE DIE BEFEHLSERWEITERUNG MIT
      SYS49260. (AUSGESCHALTET WIRD           <230>
50520 REM DIESELBE DURCH SYS49274.)           <191>
50530 REM UND WENN IHNEN DIE BEFEHLSWOR-
      TE NICHT GEFALLEN, KOENNEN              <190>
50540 REM SIE DIESE AB ZEILE 60860 OHNE
      BEDENKEN AENDERN. (ACHTEN SIE           <088>
50550 REM DARAUF, DASS SIE DIE ROUTINEN
      DURCH GOFF BZW. TEXTOFF AUS-            <234>
50560 REM SCHALTEN, INSBESONDERE DANN,
             FALLS SIE BEABSICHTIGEN, DIE
                       <025>
50570 REM DATEN ALS FILE ZU SPEICHERN.)
      ****************************            <179>
50580 REM ERLAEUTERUNGEN
      -------------                           <132>
50590 REM ZU JEDEM BEFEHL, DEN SIE NUN
      DEFINIEREN WOLLEN, GEHOEREN             <164>
50600 REM DREI DINGE:
      (1) DAS BEFEHLSWORT,                    <140>
50610 REM (2) DIE ENTSPRECHENDE MASCHI-
      NENROUTINE,                             <056>
50620 REM (3) DIE ADRESSE DIESER ROU-
      TINE.                                   <170>
50630 REM                                     <026>
50640 REM ZU (1):
      DIE BUCHSTABEN EINES BEFEHLS-           <218>
50650 REM WORTES WERDEN ALS ASCII-CODES
      GESPEICHERT, WOBEI ZU BEACH-            <191>
50660 REM TEN IST, DASS IM LETZTEN BUCH-
      STABEN DAS HOECHSTE BIT GE-             <241>
50670 REM SETZT SEIN MUSS. DIE TABELLE
      DER BEFEHLSWORTE WIRD MIT EI-           <140>
50680 REM NER NULL ABGESCHLOSSEN. AB
      ZEILE 60800 FINDEN SIE EINE             <011>
50690 REM BASIC-ROUTINE, DIE IHNEN DIE-
      SE ARBEIT ABNIMMT. ERGAENZEN            <159>
50700 REM SIE DIE TABELLE DER BEFEHLS-
      WORTE AB ZEILE 60860 MIT IH-            <071>
50710 REM REN BEFEHLEN UND ERHOEHEN SIE
      DIE VARIABLE X JE NACH ANZAHL           <118>
50720 REM DER BEFEHLE.                        <073>
50730 REM                                     <127>
50740 REM ZU (2):
      DER SPEICHERBEREICH VON $C000           <255>
50750 REM BIS EINSCHLIESSLICH ZU DER
      ADRESSE, IN DER SICH DIE AB-            <163>
50760 REM SCHLUSSNULL DER BEFEHLSWORT-
      TABELLE BEFINDET, IST FUER              <210>
50770 REM SIE TABU UND EBENSO DERJENIGE
      VON $CD00 BIS $CE41, FALLS              <111>
50780 REM SIE DIE BEFEHLE GRAFIC BIS
      TEXTOFF VERWENDEN WOLLEN. AN-           <098>
50790 REM SONSTEN BESTEHEN NUR DIE NOR-
      MALEN EINSCHRAENKUNGEN. SOL-            <116>
50792 REM LEN DIE MASCHINENPROGRAMME IN
      DATAZEILEN (AM BESTEN AB ZEI-           <053>
50794 REM LE 61700) GESPEICHERT WERDEN,
         DANN MUESSEN SIE NOCH EINEN          <084>
50796 REM ENTSPRECHENDEN BASIC-LADER AB
      ZEILE 60900 SCHREIBEN.                  <138>
```

```
50800 REM                                        <197>
50810 REM ZU (3):
      DIE ADRESSEN DER MASCHINEN-                <001>
50820 REM ROUTINEN WERDEN IN LO- UND
      HI-BYTE AUFGESPALTEN UND IN                <071>
50830 REM TABELLE DER BEFEHLSADRESSEN
      AB ZEILE 60000 EINGETRAGEN.                <078>
50840 REM DABEI IST WICHTIG,
   DASS VON             DER EIGENTLICHEN ADRES
  SE EINE                                        <253>
50850 REM EINS ABGEZOGEN WIRD!!! DAS
      EINTRAGEN DER BEFEHLSWORTE                 <043>
50860 REM UND DER BEFEHLSADRESSEN IN DIE
      ENTSPRECHENDEN TABELLEN ER-                <155>
50870 REM FOLGT PARALLEL!!! DIE ADRESSE
      VON GCLEAR ALS DRITTER BEFEHL              <085>
50880 REM Z.B. IST IN DIE 5. UND 6.
      STELLE DER TABELLE EINGETRA-               <045>
50890 REM GEN WORDEN, DAVOR BEFINDEN
      SICH DIE ADRESSEN VON GRAFIC               <152>
50900 REM UND GOFF. RECHENREGEL
   :                     (NR. DES BEFEHLS)*2 - 1 G
  IBT                                            <146>
50910 REM DIE STELLE AN, IN DIE DAS LO-
      BYTE EINGETRAGEN WIRD, IN DIE              <186>
50920 REM DARAUFFOLGENDE STELLE WIRD DAS
      HI-BYTE EINGETRAGEN. (UEBER-               <028>
50925 REM SCHREIBEN DER SICH DORT BEFIN-
      DENDEN DATEN!)                             <005>
50930 REM                                        <071>
50940 REM WENN SIE ALLE PUNKTE GENAU BE-
      ACHTET HABEN, WIRD DIE BE-                 <130>
50950 REM FEHLSERWEITERUNG, WIE OBEN
      SCHON ERWAEHNT WURDE, MIT                  <202>
50960 REM SYS(49260) INITIALISIERT.
      (AUSSCHALTEN MIT SYS(49274).)              <193>
50970 REM ******************************         <050>
50980 REM DIESE NEU DEFINIERTEN BEFEHLE
      SIND WIE BASICBEFEHLE ZU                   <243>
50990 REM HANDHABEN. SIE KOENNEN INSGE-
      SAMT 51 BEFEHLE DEFINIEREN,                <072>
51000 REM WELCHE DANN DIE CODES 204 BIS
      254 HABEN, D.H.,DASS SIE DIE-              <095>
51010 REM SE BEFEHLE AUCH POKEN KOENNEN.
      WEITERHIN KOENNEN SIE DIE BE-              <025>
51020 REM FEHLE BELIEBIG ABKUERZEN,
   WO-              BEI ALLERDINGS ZWEIDEUTIGKEI-
                           <155>
51030 REM TEN VERMIEDEN WERDEN SOLLTEN.
      BSP.: SCOLOR, CODE 207, MOEG-              <198>
51040 REM LICHE ABKUERZUNG: SC<SHIFTO>;
                      RETURN, CODE 142, MOEG-
                           <170>
51050 REM LICHE ABKUERZUNG: RET<SHIFTU>.         <195>
60000 REM                                        <217>
60005 REM *** BEFEHLSADRESSEN                    <170>
60006 REM *** $C000 - $C065                      <131>
60007 REM                                        <224>
60010 DATA 255,204,008,205,024,205,044,205,171,
   205,016,206:REM ADRESSEN DER                  <208>
60011 REM BEFEHLE GRAFIC BIS TEXTOFF             <121>
60020 DATA 182,163,182,163,182,163,182,163,182,
   163,182,163,182,163,182,163                   <029>
60030 DATA 182,163,182,163,182,163,182,163,182,
   163,182,163,182,163,182,163                   <039>
60040 DATA 182,163,182,163,182,163,182,163,182,
   163,182,163,182,163,182,163                   <049>
60050 DATA 182,163,182,163,182,163,182,163,182,
   163,182,163,182,163,182,163                   <059>
60060 DATA 182,163,182,163,182,163,182,163,182,
   163,182,163,182,163,182,163                   <069>
60070 DATA 182,163,182,163,182,163,182,163,182,
   163                                           <168>
60100 REM                                        <061>
60105 REM *** ZEIGER                             <134>
60106 REM *** $C066 - $C06B                      <000>
60107 REM                                        <068>
60110 DATA 252,192,128,192,187,192               <191>
60200 REM                                        <162>
60205 REM *** INITIALISIEREN                     <062>
60206 REM *** $C06C - $C079                      <106>
60207 REM                                        <169>
60210 DATA 120,162,005,189,102,192,157,004,003,
   202,016,247,088,096                           <057>
60300 REM                                        <006>
60305 REM *** AUSSCHALTEN                        <196>
```

```
60306 REM *** $C07A - $C07F                      <218>
60307 REM                                        <013>
60310 DATA 120,032,083,228,088,096               <126>
60400 REM                                        <106>
60405 REM *** ROUTINE (1)                        <149>
60406 REM *** $C080 - $C0BA                      <052>
60407 REM                                        <113>
60410 DATA 016,050,201,255,240,046,036,015,048,
   042,201,204,048,042,056,233,203               <055>
60420 DATA 170,132,073,160,255,234,202,240,010,
   234,200,185,205,193,016,250,048               <073>
60430 DATA 244,234,200,185,205,193,048,006,032,
   071,171,208,245,234,076,239,166               <114>
60440 DATA 234,076,243,166,234,076,036,167       <148>
60500 REM                                        <207>
60505 REM *** ROUTINE (2)                        <251>
60506 REM *** $C0BB - $C0FB                      <186>
60507 REM                                        <214>
60510 DATA 032,115,000,032,196,192,076,174,167,
   240,012,233,128,144,012,201,035               <164>
60520 DATA 176,012,076,247,167,234,076,043,168,
   234,076,165,169,234,201,075,048               <226>
60530 DATA 006,208,008,076,018,168,234,076,008,
   175,234,201,127,240,248,056,233               <214>
60540 DATA 076,010,168,185,001,192,072,185,000,
   192,072,076,115,000                           <127>
60600 REM                                        <051>
60605 REM *** ROUTINE (3)                        <096>
60606 REM *** $C0FC - $C1D1                      <017>
60607 REM                                        <058>
60610 DATA 166,122,160,004,132,015,189,000,002,
   016,007,201,255,240,065,232,208               <253>
60620 DATA 244,201,032,240,058,133,008,201,034,
   240,089,036,015,112,048,201,063               <011>
60630 DATA 208,004,169,153,208,040,201,048,144,
   004,201,060,144,032,076,150,193               <029>
60640 DATA 132,113,160,000,132,011,136,134,122,
   202,200,232,189,000,002,056,249               <011>
60650 DATA 158,160,240,245,201,128,208,048,005,
   011,164,113,232,200,153,251,001               <037>
60660 DATA 185,251,001,240,054,056,233,058,240,
   004,201,073,208,002,133,015,056               <052>
60670 DATA 233,085,208,156,133,008,189,000,002,
   240,223,197,008,240,219,200,153               <078>
60680 DATA 251,001,232,208,240,166,122,230,011,
   200,185,157,160,016,250,185,158               <079>
60690 DATA 160,208,180,189,000,002,016,190,153,
   253,001,198,123,169,255,133,122               <101>
60700 DATA 096,132,113,160,000,132,011,136,134,
   122,202,200,232,189,000,002,056               <072>
60710 DATA 249,205,193,240,245,201,128,208,008,
   005,011,024,105,076,076,074,193               <127>
60720 DATA 166,122,230,011,200,185,204,193,016,
   250,185,205,193,208,220,164,113               <122>
60730 DATA 166,122,076,047,193                   <097>
60800 REM                                        <252>
60805 REM *** BEFEHLE SPEICHERN                  <011>
60806 REM                                        <002>
60810 GOSUB 60860                                <008>
60820 Z=49613:L=0                                <252>
60830 FOR K=1 TO X:L=LEN(A$(K)):FOR I=0 TO L-2
   :POKE Z+I,ASC(MID$(A$(K),I+1,1)):NEXT       <028>
60840 POKE Z+L-1,ASC(RIGHT$(A$(K),1))OR 128
   :Z=Z+L:NEXT:POKE Z,0                          <055>
60850 PRINT"[CLEAR]":GOSUB 60900
   :PRINT"DATEN[SPACE]GESPEICHERT!"
   :PRINT"[DOWN2]SYS(49260)[HOME,DOWN]":END
                           <024>
60860 REM *** TABELLE DER BEFEHLSWORTE           <020>
60861 REM                                        <057>
60862 X=6:DIM A$(X):IF X>51 THEN END             <020>
60863 A$(1)="GRAFIC":  REM CODE 204              <175>
60864 A$(2)="GOFF":    REM CODE 205              <040>
60865 A$(3)="GCLEAR":  REM CODE 206              <183>
60866 A$(4)="SCOLOR":  REM USW.                  <087>
60867 A$(5)="GTEXT"                              <029>
60868 A$(6)="TEXTOFF"                            <179>
60890 RETURN                                     <085>
60900 REM *** SPEICHERUNG DER DATEN              <098>
60901 REM                                        <097>
60905 Z=49152                                    <231>
60906 FOR I=0 TO 460:READ X:POKE Z+I,X:NEXT
                     <056>
60910 Z=52480                                    <234>
60911 FOR I=0 TO 321:READ X:POKE Z+I,X:NEXT
                     <057>
```

```
60990 RETURN                                    <186>
61000 REM *** GRAFIC ($CD00)                    <075>
61010 DATA 169,024,160,059,162,096,076,015,205
                        <144>
61100 REM *** GOFF ($CD09)                      <046>
61110 DATA 169,021,160,027,162,169,141,024,208,
   140,017,208,142,000,205,096                  <065>
61200 REM *** GCLEAR ($CD19)                    <032>
61210 DATA 169,000,162,032,133,253,134,254,168,
   145,253,200,208,251,230,254,202              <105>
61220 DATA 208,246,096                          <195>
61300 REM *** SCOLOR ($CD2D)                    <180>
61310 DATA 201,040,240,006,076,008,175,076,072,
   178,032,155,183,224,016,176,246              <224>
61320 DATA 138,010,010,010,010,133,251,032,241,
   183,224,016,176,232,138,005,251              <190>
61330 DATA 133,251,032,247,174,174,169,205,032,
   240,233,165,251,160,039,145,209              <243>
61340 DATA 136,016,251,232,236,170,205,048,238,
   096                                          <156>
61400 REM *** GTEXT ($CD6A)                     <211>
61410 DATA 173,025,208,141,025,208,048,007,173,
   013,220,088,076,049,234,173                  <127>
61420 DATA 018,208,205,168,205,176,019,169,024,
   160,059,141,024,208,140,017,208              <078>
61430 DATA 173,168,205,141,018,208,076,188,254,
   169,021,160,027,141,024,208                  <152>
61440 DATA 140,017,208,173,167,205,141,018,208,
   076,188,254,000,255,000,025                  <147>
61441 DATA 050:REM RASTERZEILENANFANG           <021>
61445 REM EINSPRUNG ($CDAC)                     <191>
61450 DATA 240,004,201,040,240,047,120,169,106,
   141,020,003,169,205,141,021,003              <055>
61460 DATA 169,000,141,018,208,173,017,208,041,
   127,141,017,208,169,129,141,026              <110>
61470 DATA 208,169,096,141,178,205,169,120,141,
   017,206,032,121,000,088,096,076              <133>
61475 DATA 072,178,032,155,183                  <075>
61480 DATA 224,026,176,246,134,251,032,241,183,
   224,026,176,237,134,252,032,247              <138>
61490 DATA 174,166,251,228,252,176,226,162,001,
   181,251,157,169,205,010,010,010              <134>
61500 DATA 024,109,171,205,157,167,205,202,016,
   238,096                                      <000>

61600 REM *** TEXTOFF ($CE11)                   <028>
61610 DATA 096,169,049,160,234,141,020,003,140,
   021,003,169,096,141,017,206,169              <004>
61620 DATA 000,160,025,141,169,205,140,170,205,
   160,255,141,167,205,140,168,205              <255>
61630 DATA 141,026,208,169,120,141,178,205,169,
   024,032,002,205,088,096                      <148>
```

Basic auf Tastendruck

Beim Programm KEYS handelt es sich um ein Maschinenprogramm, das den Bereich $C000 - $C26B benötigt. Es erweitert das Basic um die beiden Befehle »KEY« und »DISPLAY«, mit denen die Funktionstasten belegt werden können beziehungsweise diese Belegung angezeigt werden kann. Zusätzlich können die wichtigsten Basic-Befehle durch gleichzeitiges Drücken von CTRL und einer Buchstabentaste »auf Knopfdruck« auf den Bildschirm geschrieben werden, was die Programmierarbeit doch sehr vereinfachen kann.

Bei der Initialisierung mit SYS 49152, die der Basic-Lader automatisch vornimmt, werden die Vektoren auf die Routine »Basic-Statement ausführen« und die Tastaturabfrage auf $C014 beziehungsweise $C11D verbogen.

Die beiden neuen Befehle werden an einem vorangestellten »!« erkannt, anschließend prüft das Programm auf die Anfangsbuchstaben K und D ($C014 - $C02D).Folgt keiner dieser Buchstaben, so wird ein »SYNTAX ERROR« ausgegeben ($C02E). Andernfalls verzweigt es nach $C031 (KEY) beziehungsweise $C0A7 (DISPLAY).

Der »KEY«-Befehl hat das Format !KEY N, »Befehl«. Hierbei muß N zwischen eins und zwölf liegen, der Befehl darf maximal neun Zeichen umfassen, wobei Anführungszeichen durch das Hochkomma zu ersetzen sind. Ein anschließendes RETURN erreicht man durch Eingabe von »£« als letztes Zeichen.

Die Funktionstasten F1 bis F8 werden wie üblich, die Tasten F9 bis F12 durch gleichzeitiges Drücken von »C=« und einer F-Taste erreicht.

Beispielsweise bewirkt !KEY 9, »RUN£«, daß nach gleichzeitigem Drücken von »C=« und »F1« ein Basic-Programm gestartet wird.

Der »DISPLAY«-Befehl hat das Format !DISPLAY und läßt sich ebenfalls auf eine Funktionstaste legen. Er bewirkt ein Löschen des Bildschirms und die Anzeige der Belegung der 12 Funktionstasten.

Die neue Tastaturabfrageroutine prüft zunächst, ob die Control-Taste gedrückt ist. Ist dies der Fall und ist die gleichzeitig gedrückte Taste belegt, so wird aus einer Tabelle ($C1A8-$C1DE) das Token des Befehls minus $7F geladen und mit dessen Hilfe der ASCII-Code der Buchstaben des Befehls aus der Tabelle der Basic-Befehlsworte geholt. Die einzelnen Buchstaben werden in den Tastaturpuffer geschrieben und ausgegeben.

Ist die Control-Taste nicht gedrückt, so wird geprüft, ob eine Funktionstaste gedrückt ist. Wenn ja, wird mit Hilfe des Tasten-Codes die Adresse der Belegung berechnet und die einzelnen Zeichen in den Tastaturpuffer geschrieben und ausgegeben.

(Klaus Russell/rg)

Gliederung des Programms:

$C000 - $C013	Initialisierung
$C014 - $C020	Prüfen auf »!«
$C021 - $C02D	Prüfen auf »K« oder »D«
$C02E	Ausgabe »SYNTAX ERROR«
$C031 - $C0A6	»KEY«-Befehl
$C0A7 - $C11C	»DISPLAY«-Befehl
$C11d - $C1AA	Tastaturabfrage
$C1AB - $C1DE	Tokens - $7F
$C1DF - $C1EA	[CLR/HOME] »!DISPLAY« 2 * Carriage Return
$C1EB - $C1EF	»!KEY«
$C1F0 - $C1FB	Hilfswerte zur Berechnung der Adressen der F-Tasten-Belegung
$C200 - $C26C	Belegung F-Tasten

Benutzte Routinen:

$A7E7	Basic-Statement ausführen
$AB1E	String ausgeben
$AEFD	Prüft auf Komma
$AF08	»SYNTAX ERROR« ausgeben
$B79E	Holt Byte-Wert nach X
$EB3C	$EB42, alte Tastaturabfrage
$EB48	Prüft auf SHIFT, CTRL, C=
$FFD2	Ausgabe eines Zeichens
$0073	Holt nächstes Zeichen

Belegung der Tastatur (mit Control):

Taste	Code (PEEK(203))	Befehl
+	40	SIN
-	43	COS
£	48	TAN
F1	4	RUN
F3	5	LIST
F5	6	LOAD
F7	3	SAVE
W	9	WAIT
E	14	END
R	17	RETURN
T	22	TO
Y	25	SYS
U	30	USR
I	33	INPUT
O	38	OPEN
P	41	POKE
@	46	CONT
*	49	SQR
↑	54	---
A	10	ABS
S	13	STEP
D	18	DATA
F	21	FOR
G	26	GOTO
H	29	THEN
J	34	READ
K	37	RIGHT$
L	42	LEFT$
:	45	INT
;	50	LOG
Z	12	GOSUB
X	23	PEEK
C	20	CLOSE
V	31	VAL
B	28	RESTORE
N	39	NEXT
M	36	MID$
,	47	TAB(
.	44	SPC(
CRSR↕	7	EXP
CRSR↔	2	CONT

Die Belegung läßt sich folgendermaßen ändern: Man sucht den Code der Taste, deren Belegung man ändern will, aus obiger Tabelle und addiert 49579 dazu. Nun subtrahiert man vom Token des gewünschten Befehls 127 und POKEt diesen Wert in die zuerst errechnete Adresse. Will man zum Beispiel die F1-Taste mit »VERIFY« belegen, so sieht das wie folgt aus: F1 hat den Code 4 (siehe Tabelle) Das Token für VERIFY ist 149.
Nun gibt man ein: POKE 49579+4,(149-127)
Danach ergibt <CTRL> zusammen mit F1 die Ausgabe von »VERIFY« auf dem Bildschirm. Die so geänderte Belegung läßt sich dann mit einem Monitorprogramm abspeichern. Sollen auch die Belegungen der mit !KEY belegbaren F-Tasten gespeichert werden, so muß der Bereich $C000 bis $C26B geSAVEt werden.

```
0 POKE 53280,0:POKE 53281,0:PRINT"[CLEAR]" <207>
1 PRINT"*****************************      <042>
2 PRINT"**[SPACE23]**                      <101>
3 PRINT"**[SPACE5]BASIC-BEFEHLE[SPACE5]** <224>
4 PRINT"**[SPACE4]AUF[SPACE]TASTENDRUCK[SPACE4]
  **                                       <139>
5 PRINT"**[SPACE2]+12[SPACE]FUNKTIONSTASTEN
  [SPACE2]**                               <134>
6 PRINT"**[SPACE23]**                      <105>
7 PRINT"**[SPACE6]K.[SPACE]RUSSELL[SPACE7]**
                  <013>
8 PRINT"**[SPACE6]LUISENSTR.28[SPACE5]**  <204>
9 PRINT"**[SPACE6]5100[SPACE]AACHEN[SPACE6]**
                  <210>
10 PRINT"**[SPACE23]**                          <109>
11 PRINT"*****************************          <052>
20 :                                            <078>
30 A=49152:E=49659:PRINT:PRINT                  <254>
40 PRINT"BITTE[SPACE]ETWAS[SPACE]GEDULD
   :[SPACE]ICH[SPACE]LESE[SPACE]DATEN!          <088>
60 FOR I=A TO E:READ X:S=S+X:POKE I,X:NEXT      <233>
70 IF S<>60722 THEN PRINT"DATA[SPACE]ERROR":END
                  <107>
80 FOR I=49660 TO 49771:POKE I,0:NEXT           <183>
90 SYS 49152 : REM INITIALISIERUNG              <054>
100 PRINT"[CLEAR]DIE[SPACE]FUNKTIONSTASTEN
   [SPACE]SIND[SPACE]JETZT[SPACE]3-FACH[SPACE3]
   BELEGBAR[SPACE]MIT:                          <157>
110 PRINT"[DOWN,SPACE6]!KEY[SPACE]#,
   "CHR$(34)"KOMMANDO"CHR$(34)                  <077>
120 PRINT"[DOWN]EIN[SPACE]ANSCHLIESSENDES[SPACE]
   RETURN[SPACE]IST[SPACE]MIT[SPACE]
   "CHR$(34)"£"CHR$(34)                         <246>
130 PRINT"ALS[SPACE]LETZTEM[SPACE]ZEICHEN[SPACE]
   DES[SPACE]KOMMANDOS[SPACE]ZU[SPACE]
   ER-REICHEN.                                  <108>
140 PRINT"[DOWN]DAS[SPACE]KOMMANDO[SPACE]DARF
   [SPACE]MAXIMAL[SPACE]9[SPACE]ZEICHEN[SPACE]
   UM-[SPACE]FASSEN.                            <168>
150 PRINT"[DOWN]!DISPLAY[SPACE]ZEIGT[SPACE]DIE
   [SPACE]BELEGUNG[SPACE]DER[SPACE]F-TASTENAN.
                  <017>
160 PRINT"[DOWN]DIE[SPACE]MEISTEN[SPACE]
   BASIC-BEFEHLE[SPACE]SIND[SPACE]MIT[SPACE6]
   <CTRL>[SPACE]+[SPACE]ANFANGSBUCHSTABE[SPACE]
   ";                                           <171>
170 PRINT"DES[SPACE]BEFEHLS[SPACE3]ABRUFBAR.
                  <173>
190 :                                           <248>
200 DATA 169,20,141,8,3,162,192,142,9,3,232,142,
   144,2,169,29,141,143,2,96                    <157>
201 DATA 32,115,0,201,33,240,6,32,121,0,76,231,
   167,162,0,32,115,0,201,75,240                <001>
202 DATA 7,201,68,240,121,76,8,175,32,115,0,221,
   237,193,208,245,232,224,2                    <153>
203 DATA 208,243,32,115,0,32,158,183,224,0,208,
   5,162,14,108,0,3,224,13,16                   <132>
204 DATA 247,189,239,193,133,251,169,194,133,
   252,234,234,234,32,253,174,201               <031>
205 DATA 34,240,5,162,22,108,0,3,160,255,200,
   192,9,16,29,32,115,0,201,34,240              <019>
206 DATA 39,201,39,240,11,201,92,208,9,169,13,
   145,251,76,141,192,169,34,145                <019>
207 DATA 251,76,107,192,32,115,0,201,34,240,5,
   162,23,108,0,3,192,8,16,5,200                <232>
208 DATA 169,0,145,251,162,128,108,0,3,32,115,0,
   221,226,193,208,136,232,224                  <248>
209 DATA 6,208,243,169,223,160,193,32,30,171,
   169,1,72,169,235,160,193,32,30               <017>
210 DATA 171,104,170,201,10,48,9,169,49,32,210,
   255,138,56,233,10,24,105,48                  <213>
211 DATA 32,210,255,169,44,32,210,255,169,34,32,
   210,255,189,239,193,24,105                   <230>
212 DATA 0,133,251;169,0,168,105,194,133,252,
   177,251,240,20,201,34,208,2,169              <054>
213 DATA 39,201,13,208,2,169,92,32,210,255,200,
   192,9,208,232,169,34,32,210                  <213>
214 DATA 255,169,13,32,210,255,232,138,201,13,
   208,162,240,133,173,141,2,201                <249>
215 DATA 4,240,79,162,3,228,203,240,8,232,224,7,
   208,247,76,72,235,228,197                    <193>
216 DATA 240,249,134,197,201,3,16,52,201,0,240,
   12,232,232,232,232,201,1,240                 <231>
217 DATA 4,232,232,232,232,216,169,0,224,3,240,
   8,24,105,9,202,224,3,208,248                 <254>
218 DATA 170,160,0,200,189,0,194,153,118,2,201,
   0,240,5,232,192,9,48,240,132                 <249>
219 DATA 198,76,66,235,162,2,228,203,240,7,232,
   224,51,208,247,240,175,189                   <201>
220 DATA 171,193,240,170,228,197,240,166,134,
   197,170,160,255,202,240,8,200                <235>
221 DATA 185,158,160,16,250,48,245,200,185,158,
   160,48,6,157,119,2,232,208                   <198>
222 DATA 244,56,233,128,76,60,235,255,255,27,21,
   11,28,20,62,0,19,55,0,14,42                  <015>
223 DATA 1,0,0,15,4,0,33,2,37,67,0,31,10,0,13,
   40,56,70,0,6,8,0,75,74,32,3                  <116>
224 DATA 64,24,73,63,39,54,27,36,65,59,61,255,
   147,33,68,73,83,80,76,65,89                  <239>
225 DATA 13,13,0,33,75,69,89,0,9,45,18,54,27,63,
   0,36,81,90,99,72                             <014>
```

Automatische Zeilen-numerierung

Die lästige Arbeit, vor jeder Zeile die entsprechende Nummer einzutippen, nimmt Ihnen diese kleine Routine ab.

Die Maschinen-Routine — ganze 141 Bytes — wird vom Basic-Lader über DATAs in den Speicherbereich ab Adresse $C000 beziehungsweise 49152 Dezimal geschützt abgelegt. Dabei führt das Basic-Programm eine Checksummenprüfung durch, da sich der Basic-Lader nach Ablauf von selbst löscht.

Vom Programm aus wird eine Schrittweite von 10 eingestellt. Diese kann aber durch Verändern des Wertes in Adresse 49296 — im Bereich von 1 bis 255 — eingestellt werden. Der Wert der ersten Zeilennummer wird in die Adressen: 49294 (High-Byte) und 49295 (Low-Byte) eingepoket, wenn Sie mit einem anderen Wert als 10 beginnen soll.

Beispiel: POKE 49296,S:POKE 49294,INT(Z/256):POKE 49295,Z-(INT(Z/256)*256)

S = Schrittweite ; Z = Wert der ersten Zeilennummer

Durch Eingabe von Pfeil-links/Return kann nun eine Zeilennummer angefordert werden. Die Vorgabe einer Zeilennummer wird akustisch untermalt. Nun kann eine Basic-Zeile wie gewohnt eingegeben und mit der Return-Taste abgeschlossen werden. Die nächste Ausgabe erfolgt wieder auf Anforderung, mit der eingestellten Schrittweite.

(Thomas Schulz/rg)

```
10 REM         *** AUTONUM 64 ***          <040>
11 REM                                     <154>
12 REM            AUTOMATISCHE             <034>
13 REM        ZEILENNUMMERNVORGABE         <139>
14 REM                                     <157>
15 REM                                     <158>
16 REM  1. ZEILENNR. LOW -BYTE  IN 49294   <137>
17 REM  1. ZEILENNR. HIGH-BYTE  IN 49295   <185>
18 REM  ZEILENABSTAND (1-255)   IN 49296   <085>
19 REM                                     <162>
20 REM                                     <163>
21 REM          EIN UTILITY VON:           <225>
22 REM                                     <165>
23 REM            TH.  SCHULZ              <073>
24 REM          POSTFACH 602542            <050>
25 REM          2000 HAMBURG 60            <214>
26 REM          TEL.040/6316055            <233>
27 REM                                     <170>
28 REM                                     <171>
29 POKE 49294,0:POKE 49295,0:POKE 49296,10 <194>
30 POKE 55,0:POKE 56,195:AD=49152:Z=0     <110>
31 POKE 53280,0:POKE 53281,0:PRINT CHR$(30);
                   <205>
32 PRINT CHR$(147);                        <136>
33 READ X:IF X=-1 THEN 51                  <184>
34 POKE AD,X:AD=AD+1:Z=Z+X:GOTO 33         <094>
35 DATA 32,115,0,201,95,240,6,32,121,0,76  <237>
36 DATA 231,167,32,115,0,24,173,144,192    <157>
37 DATA 109,142,192,141,142,192,144,3,238  <007>
38 DATA 143,192,162,23,160,0,24,32,10,229  <247>
39 DATA 173,143,192,174,142,192,32,205     <120>
40 DATA 189,169,152,133,251,169,7,133,252  <027>
41 DATA 169,119,133,253,169,2,133,254,160  <019>
42 DATA 0,177,251,201,32,240,6,145,253     <108>
43 DATA 200,76,66,192,132,198,160,0,162,23 <062>
44 DATA 24,32,10,229,169,15,141,24,212     <111>
45 DATA 169,207,141,7,212,169,34,141,8     <128>
46 DATA 212,169,240,141,13,212,169,17,141  <009>
47 DATA 11,212,162,64,160,255,136,208,253  <013>
48 DATA 202,208,248,169,0,141,11,212,76    <169>
49 DATA 128,164,169,0,141,8,3,169,192      <087>
50 DATA 141,9,3,96,-1                      <052>
51 IF Z<>18097 THEN 55                     <150>
52 PRINT CHR$(147)CHR$(17);                <060>
53 PRINT"AUTONUM[SPACE]64[SPACE2]ZEILENNR.VORG.
   :[SPACE2]_";                            <122>
54 PRINT CHR$(18)"RETURN":SYS 49283:NEW    <050>
55 PRINT CHR$(17)"FEHLER[SPACE]IN[SPACE]DATAS!
                   <077>
56 PRINT CHR$(17)"CHECKSUMME[SPACE3]"CHR$(18);
                   <054>
57 PRINT"[SPACE]IST:"CHR$(146)Z CHR$(18);  <009>
58 PRINT TAB(15)"SOLL:"CHR$(146)18097:END  <051>
```

Worktool — eine Programmierhilfe

Drei nützliche Optionen zur Programmeingabe werden durch dieses Programm realisiert. Farbwahl, aktuelle Zeitangabe und Warnton bei mehr als 80 Zeichen Eingabe auf Tastendruck.

Wie der Name schon sagt, hilft das »Worktool« dem Programmierer bei der Eingabe seiner eigenen Programme. Mehrere Optionen, die ansonsten nur bei erheblich teureren Geräten oder Hardware—Erweiterungen zu finden sind, werden ohne Basicspeicherverlust vom Toolkit übernommen.

Mittels Funktionstasten werden folgende Erweiterungen angesprochen:

— F1: Rahmen, Innenteil und Cursor erscheinen in einer vorher von Ihnen festgelegten Farbkombination.

Beim Schreiben von Programmen oder deren Test wird die Farbkombination, die Sie zur Eingabe verwenden, oft durch das Programm geändert. Haben Sie die Worktoolerweiterung eingegeben, können Sie sich jetzt, nach dem Probelauf 20 Tastendrücke ersparen um Farbkombinationen, wie zum Beispiel braun-rot-violett in Ihre Lieblingskombination zu verwandeln.

— F3: Ausgabe der aktuellen Uhrzeit, oder einer Stoppuhr. Wollen Sie wissen wieviel Uhr es ist, oder wie lange Sie schon mit dem Toolkit arbeiten, so genügt ein Druck auf F3. Wie sonst nur bei Steckmodulen der oberen Preisklasse wird vom Computer in der Mitte der 1. Zeile die Zeit in reverser Darstellung ausgegeben. Jedoch nicht permanent, sondern nur solange die Taste gedrückt ist. Das hat den Vorteil, daß keine Bildschirmzeile verloren geht.

— F5 und F7: Falls der eine oder andere von Ihnen schon einmal am CBM 4032 gearbeitet hat, wird er bestimmt bemerkt haben, daß dieser eine praktische Eingabehilfe besitzt. Er gibt, wenn der Cursor eine bestimmte Spalte erreicht, einen Piepser aus. Diese Option wurde beim Worktool noch erheblich verbessert. Im Direktmodus gibt er jetzt immer einen hellen Ton (ähnlich Tastaturklick) aus, sobald der Cursor die Spalte 35 oder 75 erreicht. Im Programmeingabemodus dagegen wird bei 35 der Klick, bei 75 ein Sägezahnton, und bei 80 ein auf-

und abschwellender Ton erzeugt. Jeder kennt das ungeheuer langwierige und nervenzermürbende Editieren einer falsch eingegebenen PRINT-Zeile. Gerade im Programmierrausch schreibt man oft über die maximale Zeilenlänge hinaus, und muß später zeitaufwendig ausbessern. Ist mit F5 die Tonoption eingeschaltet, gibt der Computer rechtzeitig die oben genannten »Erinnerungstöne« aus. Mit F7 läßt sich diese Funktion ausschalten (hebt das Klima beim Mittagsschlaf Ihrer Frau).

Im Programm selbst wurde weitgehend auf Steuerzeichen verzichtet, und stattdessen der CHR$ und POKE-Befehl verwendet. Die DATAs sind der Übersicht und der Arbeitsersparnis halber in hexadezimaler Form einzugeben und werden von einer Routine automatisch umgerechnet. Das Programm gibt bei Fehleingabe einen Checksumfehler mit der Angabe des betreffenden Datenblocks aus. Die Funktionen einzelner Abschnitte sind in REM-Zeilen kurz umrissen. Das Basic-Programm benötigt 3k Speicher. Nachdem Sie Ihre Eingaben gemacht haben, löscht sich der Basic-Teil selbst und nur noch zirka 250 Bytes im abgeteilten Maschinensprachespeicher bei $C000 sind nötig. Es ist also der gesamte Basic-RAM frei verfügbar.

(Ulrich Grothaus / rg)

Zeilen	Beschreibung
10-110	Ausgabe von Kopf und Menü
120-140	Eingabe der gewünschten Farbkombinationen
220	Farbwerte für Maschinenprogramm merken
230	Eingabe der aktuellen Uhrzeit beziehungsweise wenn nur RETURN der Nullstellung der Stoppuhr
240	diese Funktion rechnet den Dezimalwert in den TOD-BCD-Wert um
250-280	Auswerten des Zeitstrings, Einlesen der Werte in die CIA's und initialisieren der TOD Register, sowie starten der Uhr
260-410	Einlesen der Vorroutine und des Hauptteils sowie Start der Vorroutine. Die GOSUBs springen zum Umrechnen und Einpoketeil
490	Mit dem POKE 792, 193 wird das Lowbyte des NMI-Interruptvektors auf einen RTI gelenkt. Dadurch ist die RESTORE-Taste ausgeschaltet, was ein Zurücksetzen der unteren Pages verhindert. Im Bedarfsfall könnten Sie dies durch Löschen der Zeile oder nachträglich mit POKE 792,71 rückgängig machen.
500-590	Kurze Erklärungen und Abfrage, ob sich das Basic-Programm löschen kann. Geben Sie »N« ein, beginnt es von vorne.
660-690	Ausgabe des Kopfes
700-710	Sobald eine Taste gedrückt wird, löscht sich der Bildschirm, der Basic-Teil, und das Maschinenprogramm wird gestartet. Ferner wird der Bildschirm gelöscht, so daß der gesamte Bildschirm zum Programmieren frei ist und kein Ausgabekopf stört.
710	Die Schleife dient auf recht unkonventionelle Weise dazu, das Basic-Programm zu löschen, aber trotzdem noch den SYS abzuarbeiten.
720-760	Einleseroutine für den 1. Maschinensprachteil. Zeile 730 und 740 rechnen die hexadezimalen DATAs in Dezimalzahlen von 0 bis 255 um.
770-790	Diese DATA-Zeilen enthalten den 1. Teil des Maschinenprogramms.
800-840	Einleseroutine für den 2. Maschinensprachteil.
850-1180	Diese DATA-Zeilen enthalten das Interruptprogramm.

Programmbeschreibung

Adresse	Beschreibung
$ C000-C00D	Der Interrruptvektor wird auf das Worktool zusätzlich zur normalen Interruptroutine des Betriebssystems jede sechzigstel Sekunde abgearbeitet wird.
$ C00E-C03A	Hier wird entschieden ob aufgrund einer gedrückten Funktionstaste oder eines Registerwertes in ein Unterprogramm verzweigt werden soll. Am Ende dieses Teils beziehungsweise einer aufgerufenen Unterroutine wird die normale Interruptroutine angesprungen.
$C040-C052	Falls F1 gedrückt ist, setzt diese Routine die Farbregister auf den eingestellten Wert, das heißt der Bildschirm erscheint in den von Ihnen gewünschten Farben.
$ C057-C08E	Wenn F3 gedrückt ist, werden die TOD-Register ausgewertet und die Zeit aktualisiert beziehungsweise ausgegeben. Dies hat gegenüber ähnlichen Programmen den Vorteil daß Sie die Zeit bei ihrer Programmeingabe überschreiben können, was bei der Ausgabe der Zeit im Interrupt nicht möglich gewesen wäre.Sie könnten ansonsten die 1. Bildschirmzeilen nicht benutzen, weil das einen Syntax Error hervorrufen würde.
$ C091-C0A6	Die derzeitige Cursorposition wird abgefragt und bei Bedarf, sofern Sie diese Option aktiviert haben, in eine Tonausgaberoutine gesprungen.
$ C0AD-C109	In diesem Teil werden die verschiedenen Töne ausgegeben.

Aufbau der Maschinenroutine

```
0 REMWORKTOOL                                   <116>
1 REMULRICH GROTHAUS                            <040>
2 REMFALLWEG 18                                 <096>
3 REM8450 AMBERG                                <117>
4 :                                                  <062>
5 :                                                  <063>
10 REM ------------------------------------     <057>
20 REM       ABFRAGE DER OPTIONSWERTE           <025>
30 REM ------------------------------------     <077>
40 :                                                 <098>
50 :                                                 <108>
60 RESTORE:POKE 53280,0:POKE 53281,0            <039>
70 PRINT CHR$(147):POKE 646,5
   :PRINT"******************[WHITE]WORKTOOL[GREEN]
   ********************"                        <096>
80 POKE 646,7:PRINT TAB(8)"VON[SPACE]ULRICH
   [SPACE]GROTHAUS"                             <044>
90 POKE 646,15:PRINT:PRINT:PRINT:PRINT"OPTIONS
   :";,:POKE 646,10                             <126>
100 PRINT"F1=FARBE[SPACE]SETZEN"
   :PRINT,"F3=ZEIT[SPACE]AUSGEBEN"
   :PRINT,"F5=WARNTON[SPACE]EIN"                <173>
110 PRINT,"F7=WARNTON[SPACE]AUS":PRINT:PRINT
   :POKE 646,14                                 <235>
120 INPUT"FARBCODE[SPACE]RAHMEN";R              <191>
130 INPUT"FARBCODE[SPACE]GRUND";G               <131>
140 INPUT"FARBCODE[SPACE]CURSOR";C              <231>
150 :                                           <208>
160 :                                           <218>
```

```
170 REM --------------------------------     <172>
180 REM WERTUEBERGABE MASCHINENPROGRAMM      <002>
190 REM --------------------------------     <192>
200 :                                        <002>
210 :                                        <012>
220 POKE 49425,R:POKE 49426,G:POKE 49427,C <144>
230 PRINT:PRINT TAB(7);"HHMMSS":PRINT CHR$(145);
   :INPUT"ZEIT[SPACE]";Z$                    <092>
240 DEF FN F(X)=INT(X/10)*16+(X-INT(X/10)*10)
                   <194>
250 H=VAL(LEFT$(Z$,2)):M=VAL(MID$(Z$,3,2))
   :S=VAL(RIGHT$(Z$,2))                      <239>
260 POKE 56335,PEEK(56335)AND 127            <048>
270 POKE 56329,FN F(S):POKE 56330,FN F(M)
   :POKE 56331,FN F(H):POKE 56328,0          <169>
280 POKE 56334,PEEK(56334)OR 128             <068>
290 :                                        <093>
300 :                                        <103>
310 REM --------------------------------     <057>
320 REM        EINLESEN DER DATEN            <106>
330 REM --------------------------------     <077>
340 :                                        <143>
350 :                                        <153>
360 GOSUB 720                                <143>
370 FOR T=1 TO 4:SYS 36500:NEXT              <149>
380 FOR T=1 TO 40:PRINT:NEXT                 <204>
390 POKE 646,13:PRINT"DAS[SPACE2]EINLESEN
   [SPACE2]DES[SPACE]MASCHINENPROGRAMMES[SPACE]
   DAUERT[SPACE]15[SPACE]SEKUNDEN"           <135>
400 GOSUB 800                                <182>
410 FOR T=1 TO 4:SYS 36500:NEXT              <189>
420 :                                        <223>
430 :                                        <233>
440 REM --------------------------------     <142>
450 REM       BENUTZERHINWEISE               <029>
460 REM --------------------------------     <162>
470 :                                        <017>
480 :                                        <027>
490 POKE 792,193:PRINT CHR$(147):POKE 646,7
                   <072>

500 PRINT"AUS[SPACE]PROGRAMMTECHNISCHEN[SPACE]
   GRUENDEN[SPACE]IST[SPACE]DER[SPACE,WHITE]
   NMI-INTERRUPT";:POKE 646,7                <097>
510 PRINT"[SPACE]AUSGESCHALTET(ZEILE[SPACE]
   490)."                                    <062>
520 PRINT"DAS[SPACE]WORKTOOL[SPACE]LIEGT[SPACE]
   BEI[SPACE]49152."                         <184>
530 PRINT"ES[SPACE]SIND[SPACE]38911[SPACE]BYTES
   [SPACE]BASIC-RAM[SPACE]FREE."             <003>
540 POKE 53280,0:POKE 646,3:PRINT"[DOWN14]KANN
   [SPACE]ICH[SPACE]MICH[SPACE]LOESCHEN[SPACE]
   (J/N) ?"                                  <077>
550 FOR T=1 TO 11:GET N$:NEXT                <239>
560 GET A$:IF A$=""THEN 560                  <154>
570 IF A$="N"THEN GOTO 60                    <006>
580 :                                        <128>
590 :                                        <138>
600 REM --------------------------------     <092>
610 REM     LOESCHEN DES BASICTEILS          <003>
620 REM    START DER TOOL-IRQ ROUTINE        <227>
630 REM --------------------------------     <122>
640 :                                        <188>
650 :                                        <198>
660 POKE 53280,PEEK(49425):POKE 53281,
   PEEK(49426):POKE 646,PEEK(49427)          <076>
670 PRINT CHR$(147);"******************@WORKTOOL@
   ****************"                         <123>
680 PRINT"[SPACE7]-----38634[SPACE]BYTES[SPACE]
   FREE----":PRINT                           <160>
690 PRINT"**************1984[SPACE]BY[SPACE]
   GROTO***************"                     <251>
700 GET A$:IF A$=""THEN 700                  <034>
710 PRINT CHR$(147):FOR T=2048 TO 2060:POKE T,0
   :NEXT:POKE 45,3:POKE 46,8:SYS 49152:END <028>
720 PS=0:FOR AD= 36500 TO 36519 :READ DA$    <093>
730 L=ASC(LEFT$(DA$,1))-48:R=ASC(RIGHT$(DA$,
   1))-48                                    <179>
740 PO=(L+(L>9)*7)*16+R+(R>9)*7:POKE AD,PO
   :PS=PS+PO:NEXT                            <105>
750 IF PS<> 2456 THEN PRINT"[CLEAR]
   PRUEFSUMMENFEHLER[SPACE]IM[SPACE]1.[SPACE]
   DATABLOCK.":END                           <217>
760 RETURN                                   <136>
770 DATA A0,00,A2,00,8E,20,D0,E8             <048>
780 DATA F0,03,4C,98,8E,C8,F0,03             <088>
790 DATA 4C,96,8E,60                         <233>
800 PS=0:FOR AD= 49152 TO 49420 :READ DA$  <144>
810 L=ASC(LEFT$(DA$,1))-48:R=ASC(RIGHT$(DA$,
   1))-48                                    <004>
820 PO=(L+(L>9)*7)*16+R+(R>9)*7:POKE AD,PO
   :PS=PS+PO:NEXT                            <186>
830 IF PS<> 32697 THEN PRINT"[CLEAR]
   PRUEFSUMMENFEHLER[SPACE]IM[SPACE]2.[SPACE]
   DATABLOCK.":END                           <101>
840 RETURN                                   <217>
850 DATA EA,78,A9,0E,8D,14,03,A9             <174>
860 DATA C0,8D,15,03,58,60,8D,04             <129>
870 DATA D4,EA,A9,04,C5,C5,F0,28             <204>
880 DATA A9,05,C5,C5,F0,39,A9,06             <187>
890 DATA C5,C5,D0,03,8D,10,C1,A9             <198>
900 DATA 03,C5,C5,D0,03,8D,10,C1             <185>
910 DATA AD,10,C1,C9,06,F0,5A,4C             <231>
920 DATA 31,EA,60,00,EA,EA,EA,EA             <026>
930 DATA AD,11,C1,8D,20,D0,AD,12             <241>
940 DATA C1,8D,21,D0,AD,13,C1,8D             <006>
950 DATA 86,02,4C,18,C0,EA,EA,AD             <036>
960 DATA 0B,DC,29,7F,A2,10,20,78             <000>
970 DATA C0,AD,0A,DC,A2,13,20,78             <028>
980 DATA C0,AD,09,DC,A2,16,20,78             <033>
990 DATA C0,AD,08,DC,4C,31,EA,00             <064>
1000 DATA 48,29,F0,4A,4A,4A,4A,18            <044>
1010 DATA 69,B0,9D,00,04,68,29,0F            <029>
1020 DATA 18,69,B0,9D,01,04,60,EA            <046>
1030 DATA EA,A5,D3,C9,23,F0,11,A5            <099>
1040 DATA D3,C9,46,F0,2F,A5,D3,C9            <127>
1050 DATA 4F,F0,4D,4C,31,EA,60,00            <100>
1060 DATA EA,EA,EA,A9,06,8D,00,D4            <165>
1070 DATA A9,78,8D,01,D4,A9,0A,8D            <142>
1080 DATA 18,D4,A9,33,8D,05,D4,A9            <135>
1090 DATA 00,8D,06,D4,A9,11,8D,04            <115>
1100 DATA D4,4C,31,EA,EA,EA,EA,A9            <236>
1110 DATA 06,8D,00,D4,A9,32,8D,01            <135>
1120 DATA D4,A9,0A,8D,18,D4,A9,63            <190>
1130 DATA 8D,05,D4,A9,00,8D,06,D4            <178>
1140 DATA A9,21,8D,04,D4,4C,31,EA            <197>
1150 DATA A9,01,8D,03,D4,A9,0F,8D            <215>
1160 DATA 18,D4,A9,FF,8D,06,D4,A9            <254>
1170 DATA 00,8D,05,D4,A9,41,8D,04            <197>
1180 DATA D4,4C,31,EA,00                     <012>
```

Mini-GBasic für den VC 20

Dieses Programm für VC 20 mit mindestens 16 KByte-Speichererweiterung implementiert zehn neue Befehle, die sich auf die Bildschirmdarstellung und die Grafik des VC 20 beziehen.

Mini-GBasic ist eine Befehlserweiterung, die schnelle Grafik-Routinen in Maschinensprache beinhaltet. Mini-GBasic bietet 10 neue Befehle zum Beispiel zum Setzen oder Löschen eines Punktes oder zum Verbinden zweier beliebiger Punkte. Außerdem wird ein Scrollen des Bildschirmes in alle vier Richtungen ermöglicht, was bei Spielen sehr nützlich sein kann. Mini-GBasic zeichnet sich dadurch aus, daß die neuen Befehle voll implementiert sind, so daß auch nach einem IF...THEN-Statement kein Doppelpunkt gesetzt werden muß, wenn einer der neuen Befehle folgt (dies ist bei, mir aus anderen Computerzeitschriften bekannten, ähnlichen Programmen leider nicht der Fall).

Zu erwähnen ist noch, daß man bei Mini-GBasic für die Befehle DRAW beziehungsweise CDRAW und SET beziehungsweise RESET benötigte Parameter, wie bei Basic-Funktionen, in Klammern eingeschlossen übergibt (dabei wird zur Übernahme beliebiger Ausdrücke das ROM-Unterprogramm GETBYTE ab Adresse $ D79B verwendet).

Zur Fehlerbehandlung

Das Programm gibt die im »normalen« Basic vorhandenen Fehlermeldungen »illegal quantity error in...« bei für die Grafik-Befehle nicht definierten Parametern und »syntax error in...« bei fehlenden Klammern und falsch eingegebenen Befehlsworten, die immer ausgeschrieben werden müssen, aus.

Zu den allgemeinen Möglichkeiten des Programmes ist noch zu sagen, daß es mit 29568 (x-Achse:0-167; y-Achse:0-175) im 16x8-Byte-Modus arbeitet.

Wichtig: Falls Sie das mit dem »Basic-Lader« geSAVEte Maschinenprogramm testen wollen, müssen Sie nach dem Ladevorgang erst einen System-Reset mit SYS 64802 durchführen und danach das eigentliche Maschinenprogramm mit SYS 9794 starten (das Programm setzt automatisch den Basic-Start herauf und führt den Befehl »New« aus).

Zur Eingabe des Programms

- Vor dem Eintippen des Basic-Laders wird der Basic-Start mit den Befehlen POKE 9823,0:POKE 43,96:POKE 44,38:NEW heraufgesetzt. Dies ist notwendig, da sich der Basic-Lader sonst selber überschreiben würde.
- Nach dem Start des Basic-Laders wird das Maschinenprogramm eingePOKEt und eine Prüfsumme erstellt. Bei fehlerhafter Eingabe der DATAs wird eine Fehlermeldung ausgegeben.
- Der Basic-Lader wartet mit Ausgabe der Frage »Bereit zum Absaven?« auf Eingabe der Taste »j«. Danach wird das Maschinenprogramm abgeSAVEt.
- Das nach der eben beschriebenen Vorgehensweise geSAVEte Maschinenprogramm kann nun einfach durch Eingabe von »LOAD« an die Stelle geladen werden, an der es vor dem AbSAVEn gestanden hat. Dies wird dadurch erreicht, daß es vorher mit dem Befehl »SAVE "MINI GBASIC", 1,1« geSAVEt wurde.
- Nach dem Einladen des Maschinenprogrammes muß ein System-Reset durchgeführt werden (SYS 64802), wonach die Befehlserweiterung mit SYS 9794 gestartet werden kann. Dieser SYS-Befehl muß auch nach eventuell durchgeführten System-Resets eingegeben werden, da dabei alle Basic-Vektoren zurückgesetzt werden. Nachdem sich die Befehlserweiterung mit »READY« gemeldet hat, können die Befehle benutzt werden.

Vor allen Befehlen muß ein Ausrufezeichen, das Erkennungsmerkmal für den Computer, daß jetzt einer der neuen Befehle folgt, stehen, da sonst eine »Syntax error«-Meldung ausgegeben wird. Nach dem Ausrufezeichen folgt der neue Befehl voll ausgeschrieben.

Die Befehle im einzelnen

!GMODE: Umschalten in den Grafikmodus und Löschen des Zeichensatzes.
!TMODE: Umschalten in den Textmodus und Löschen des Bildschirmes.
!DSCROLL: Scrollt den Bildschirm (im Textmodus) um eine Zeichenposition nach unten.
!USCROLL: Scrollt den Bildschirm (im Textmodus) um eine Zeichenposition nach oben.
!LSCROLL: Scrollt den Bildschirm (im Textmodus) um eine Zeichenpostition nach links.
!RSCROLL: Scrollt den Bildschirm (im Textmodus) um eine Zeichenposition nach rechts.
!SET (x,y): Setzt einen Punkt (im Grafikmodus) auf die Koordinate (x,y).
!RESET (x,y): Löscht einen Punkt (im Grafikmodus) auf der Koordinate (x,y,).
!DRAW (x,y): Zieht eine Linie (im Grafikmodus) von der letzten durch !SET,!RESET,!DRAW oder !CDRAW gesetzten Koordinate zur Koordinate (x,y).
!CDRAW (x,y): Löscht eine Linie (im Grafikmodus) von der letzten durch !SET,!RESET,!DRAW oder !CDRAW gesetzten Koordinate zur Koordinate (x,y)
(x,y): x = beliebiger numerischer-oder String-Ausdruck mit dem Wert 0-167
y = beliebiger numerischer-oder String-Ausdruck mit dem Wert 0-175
!CLEAR: Löscht den Grafik-Bildschirm

Anmerkung zu !GMODE:

Die Farbe, in der die Punkte auf den Bildschirm gebracht werden, läßt sich verändern, indem man den Inhalt der Speicher-

Tabelle 1. Übergabe der Parameter für SET+RESET und DRAW+CDRAW:

SET+RESET: x1,y1
DRAW+CDRAW: x2,y2

Adresse	Variable
$033C	x1
$033D	y1
$0359	x2
$035A	y2

Tabelle 2. Routinen und DATA-Felder beim Mini-GBasic

Adresse in HEX	
$1100-2000	Zeichenspeicher
$2000	GMODE
$2028	CLEAR
$2050	TMODE
$20CA	SET+RESET
$21CO	DSCROLL
$2250	LSCROLL
$22AO	RSCROLL
$22E5	INITIAL
$2300	sucht Ausrufezeichen im Basic-Text
$2312	DECODE
$2378	neue IF-THEN Routine
$23BA-23FF	DATA: Befehlswort
$25BA-25BF	DATA: Adressen der Routinen

zelle 8224 (dezimal) mit dem gewünschten Farbcode belegt, bevor der Befehl !GMODE, der dann den Farbcode in den Farbspeicher bringt, ausgeführt wird.

Vorsicht ist bei der Benutzung der Grafik im Direktmodus des Computers geboten. Da der neue Zeichensatz teilweise im Bildschirm-RAM-Bereich liegt (das ist notwendig um mit der hohen Auflösung von 168 x 176 Punkten arbeiten zu können), ist es möglich, mit dem Cursor in den ersten 32 neu definierten 16 x 8 Bit Zeichen »herumzufahren«.

Für diejenigen, die selber in Maschinensprache programmieren, ist hier noch eine Liste mit den Einsprungadressen der wichtigsten Routinen und der Lage der Befehlswort-DATAs abgedruckt (Tabellen 1 und 2). Zur Benutzung der Routinen SET/RESET und DRAW/CDRAW sei noch angemerkt, daß ein Einsprung in die jeweilige Routine mit gesetztem Carry-Flag das Setzen eines Punktes beziehungsweise das Zeichnen einer Linie und der Einsprung mit gelöschtem Carry-Flag das Löschen eines Punktes oder einer Linie bewirkt.

(Jürgen Skerhut / ev)

```
0 REM  MINI GBASIC              <101>
1 REM                           <144>
2 REM JUERGEN SKERHUT           <199>
3 REM DONATUSSTR.5              <012>
4 REM 5210 TROISDORF            <023>
5 REM                           <148>
6 REM                           <149>
7 REM                           <150>
22 DATA 32,95,229,169           <045>
24 DATA 151,141,3,144           <024>
26 DATA 169,21,141              <142>
28 DATA 2,144,169,204           <036>
30 DATA 141,5,144,169           <041>
32 DATA 14,141,0,144            <203>
34 DATA 162,0,160,16            <240>
36 DATA 152,157,0,16            <247>
38 DATA 169,6,157,0             <208>
40 DATA 148,232,200             <201>
42 DATA 208,243,169             <216>
44 DATA 17,160,0,133            <249>
46 DATA 1,132,0,152             <196>
48 DATA 145,0,24,169            <007>
50 DATA 1,101,0,133             <195>
52 DATA 0,169,0,101             <205>
54 DATA 1,133,1,201             <201>
56 DATA 32,208,236              <173>
58 DATA 76,174,199              <193>
60 DATA 234,234,234             <226>
62 DATA 234,234,234             <228>
64 DATA 234,234,234             <230>
66 DATA 169,12,141              <182>
68 DATA 0,144,169,22            <024>
70 DATA 141,2,144,169           <078>
72 DATA 174,141,3,144           <077>
74 DATA 169,192,141             <247>
76 DATA 5,144,32,95             <244>
78 DATA 229,96,32,80            <046>
80 DATA 32,76,174,199           <104>
82 DATA 234,234,169             <255>
84 DATA 0,141,65,3              <191>
86 DATA 141,66,3,141            <040>
88 DATA 67,3,162,8              <208>
90 DATA 78,64,3,144             <006>
92 DATA 19,24,173,62            <054>
94 DATA 3,109,66,3              <209>
96 DATA 141,66,3,173            <055>
98 DATA 63,3,109,67             <012>
100 DATA 3,141,67,3             <212>
102 DATA 78,67,3,110            <014>
104 DATA 66,3,110,65            <011>
106 DATA 3,202,208              <166>
108 DATA 220,96,234             <227>
110 DATA 234,234,56             <230>
112 DATA 173,68,3,237           <079>
114 DATA 69,3,141,70            <024>
116 DATA 3,96,24,173            <030>
118 DATA 71,3,101,0             <218>
120 DATA 133,0,173              <181>
122 DATA 72,3,101,1             <224>
124 DATA 133,1,96,169           <090>
126 DATA 0,133,0,133            <019>
128 DATA 1,96,234,234           <089>
130 DATA 234,8,169              <206>
132 DATA 0,162,0,160            <027>
134 DATA 0,32,192,32            <035>
136 DATA 169,80,141             <001>
138 DATA 62,3,169,1             <253>
140 DATA 141,63,3,173           <096>
142 DATA 61,3,41,240            <042>
144 DATA 74,74,74,74            <067>
146 DATA 141,64,3,32            <049>
148 DATA 112,32,173             <003>
150 DATA 65,3,141,71            <057>
152 DATA 3,173,66,3             <012>
154 DATA 141,72,3,32            <056>
156 DATA 176,32,169             <026>
158 DATA 16,141,64              <224>
160 DATA 3,173,60,3             <014>
162 DATA 41,248,74              <235>
164 DATA 74,74,141              <235>
166 DATA 62,3,169,0             <024>
168 DATA 141,63,3,32            <070>
170 DATA 112,32,173             <025>
172 DATA 65,3,141,71            <079>
174 DATA 3,173,66,3             <034>
176 DATA 141,72,3,32            <078>
178 DATA 176,32,173             <043>
180 DATA 61,3,41,15             <032>
182 DATA 141,71,3,169           <142>
184 DATA 0,141,72,3             <033>
186 DATA 32,176,32              <253>
188 DATA 169,7,141              <004>
190 DATA 68,3,173,60            <103>
192 DATA 3,41,7,141             <044>
194 DATA 69,3,32,165            <108>
196 DATA 32,169,17              <012>
```

```
198 DATA 141,72,3,169        <159>
200 DATA 0,141,71,3          <048>
202 DATA 32,176,32           <013>
204 DATA 174,70,3,189        <171>
206 DATA 117,33,141          <062>
208 DATA 73,3,165,1          <065>
210 DATA 201,32,144          <062>
212 DATA 2,40,96,160         <119>
214 DATA 0,173,73,3          <069>
216 DATA 40,176,7,73         <130>
218 DATA 255,49,0,145        <180>
220 DATA 0,96,17,0           <026>
222 DATA 145,0,96,1          <079>
224 DATA 2,4,8,16,32         <125>
226 DATA 64,128,234          <091>
228 DATA 234,169,90          <097>
230 DATA 133,43,169          <095>
232 DATA 38,133,44           <045>
234 DATA 169,0,141           <043>
236 DATA 89,38,32,68         <162>
238 DATA 198,96,165          <118>
240 DATA 159,133,164         <156>
242 DATA 165,1,133           <049>
244 DATA 166,165,158         <166>
246 DATA 133,163,133         <153>
248 DATA 165,56,233          <114>
250 DATA 22,133,158          <110>
252 DATA 133,0,165           <058>
254 DATA 159,233,0           <064>
256 DATA 133,159,24          <120>
258 DATA 105,132,133         <161>
260 DATA 1,96,234,234        <222>
262 DATA 234,234,169         <180>
264 DATA 32,160,21           <067>
266 DATA 145,158,136         <184>
268 DATA 16,251,96           <086>
270 DATA 234,169,17          <139>
272 DATA 133,159,133         <185>
274 DATA 164,169,149         <199>
276 DATA 133,1,133           <079>
278 DATA 166,169,206         <199>
280 DATA 133,158,133         <192>
282 DATA 0,169,228           <098>
284 DATA 133,163,133         <192>
286 DATA 165,234,160         <198>
288 DATA 21,177,158          <124>
290 DATA 145,163,177         <209>
292 DATA 0,145,165           <102>
294 DATA 136,16,245          <158>
296 DATA 32,144,33           <104>
298 DATA 165,159,201         <212>
300 DATA 15,208,234          <161>
302 DATA 169,16,133          <168>
304 DATA 159,169,0           <123>
306 DATA 133,158,32          <168>
308 DATA 181,33,76           <125>
310 DATA 174,199,234         <234>
312 DATA 234,234,8           <126>
314 DATA 24,165,158          <182>
316 DATA 105,22,133          <169>
318 DATA 158,133,0           <127>
320 DATA 169,0,101           <126>
322 DATA 159,133,159         <243>
324 DATA 133,164,24          <184>
326 DATA 169,132,101         <234>
328 DATA 159,133,1           <139>
330 DATA 133,166,40          <190>
332 DATA 144,8,164           <148>
334 DATA 158,136,132         <248>
336 DATA 163,132,165         <248>
338 DATA 96,164,158          <214>
340 DATA 200,132,163         <242>
342 DATA 132,165,96          <211>
344 DATA 234,234,160         <253>
346 DATA 0,162,32,138        <044>
348 DATA 145,158,165         <012>
350 DATA 158,197,163         <019>
352 DATA 240,16,24           <159>
354 DATA 169,22,101          <212>
356 DATA 158,133,158         <019>
358 DATA 169,0,101           <164>
360 DATA 159,133,159         <025>
362 DATA 76,52,34,96         <028>
364 DATA 234,234,169         <026>
366 DATA 16,133,159          <231>
368 DATA 133,164,169         <030>
370 DATA 148,133,1           <179>
372 DATA 133,166,169         <036>
374 DATA 0,133,163           <179>
376 DATA 133,165,169         <039>
378 DATA 1,133,158           <188>
380 DATA 133,0,234           <184>
382 DATA 160,0,177           <192>
384 DATA 158,145,163         <046>
386 DATA 177,0,145           <199>
388 DATA 165,200,192         <042>
390 DATA 21,208,243          <248>
392 DATA 165,163,201         <045>
394 DATA 228,240,7           <207>
396 DATA 56,32,0,34          <251>
398 DATA 76,105,34           <212>
400 DATA 169,16,133          <010>
402 DATA 159,169,21          <016>
404 DATA 133,158,169         <069>
406 DATA 249,133,163         <066>
408 DATA 32,48,34,76         <069>
410 DATA 174,199,32          <026>
412 DATA 117,233,76          <022>
414 DATA 174,199,234         <082>
416 DATA 234,234,169         <078>
418 DATA 16,133,159          <027>
420 DATA 133,164,169         <082>
422 DATA 148,133,1           <231>
424 DATA 133,166,169         <088>
426 DATA 0,133,158           <235>
428 DATA 133,0,169           <239>
430 DATA 1,133,163           <236>
432 DATA 133,165,234         <088>
434 DATA 160,20,177          <038>
436 DATA 158,145,163         <098>
438 DATA 177,0,145           <251>
440 DATA 165,136,16          <049>
442 DATA 245,165,158         <107>
444 DATA 201,228,240         <093>
446 DATA 7,24,32,0           <248>
448 DATA 34,76,185           <014>
450 DATA 34,169,16           <012>
452 DATA 133,159,169         <118>
454 DATA 0,133,158           <007>
456 DATA 169,228,133         <119>
458 DATA 163,32,48           <017>
460 DATA 34,76,174           <024>
462 DATA 199,169,0           <029>
464 DATA 141,8,3,141         <111>
466 DATA 60,3,141,61         <112>
468 DATA 3,169,35,141        <173>
```

```
470 DATA 9,3,141,1          <017>
472 DATA 3,169,169          <039>
474 DATA 141,0,3,76         <072>
476 DATA 127,33,32          <029>
478 DATA 115,0,201          <020>
480 DATA 139,240,110        <129>
482 DATA 201,33,240         <077>
484 DATA 7,32,121,0         <076>
486 DATA 76,231,199         <104>
488 DATA 234,162,0          <038>
490 DATA 142,95,3,32        <143>
492 DATA 115,0,201          <034>
494 DATA 65,144,18          <055>
496 DATA 157,96,3,232       <206>
498 DATA 224,8,208          <056>
500 DATA 241,234,162        <153>
502 DATA 11,76,55,196       <215>
504 DATA 234,234,234        <159>
506 DATA 234,142,94         <115>
508 DATA 3,160,0,162        <151>
510 DATA 0,189,96,3         <122>
512 DATA 217,186,35         <126>
514 DATA 208,10,232         <113>
516 DATA 200,236,94         <123>
518 DATA 3,144,241          <070>
520 DATA 76,97,35,238       <243>
522 DATA 95,3,173,95        <189>
524 DATA 3,201,11,144       <214>
526 DATA 3,76,39,35         <139>
528 DATA 185,186,35         <146>
530 DATA 200,201,33         <122>
532 DATA 208,248,76         <150>
534 DATA 53,35,173          <094>
536 DATA 95,3,10,168        <194>
538 DATA 185,186,37         <158>
540 DATA 141,0,1,185        <186>
542 DATA 187,37,141         <155>
544 DATA 1,1,108,0          <052>
546 DATA 1,32,115,0         <136>
548 DATA 169,199,72         <173>
550 DATA 169,174,72         <168>
552 DATA 32,158,205         <159>
554 DATA 32,121,0,201       <239>
556 DATA 137,240,3          <109>
558 DATA 169,167,32         <174>
560 DATA 255,206,165        <221>
562 DATA 97,208,3,32        <221>
564 DATA 9,201,240          <115>
566 DATA 14,32,121          <113>
568 DATA 0,176,1,76         <173>
570 DATA 160,200,104        <213>
572 DATA 104,32,121         <167>
574 DATA 0,76,3,35          <127>
576 DATA 76,251,200         <180>
578 DATA 138,72,173         <191>
580 DATA 5,144,201          <130>
582 DATA 204,208,3          <134>
584 DATA 32,80,32,104       <024>
586 DATA 170,76,58          <153>
588 DATA 196,71,77          <159>
590 DATA 79,68,69,33        <010>
592 DATA 84,77,79,68        <017>
594 DATA 69,33,68,83        <009>
596 DATA 67,82,79,76        <017>
598 DATA 76,33,85,83        <010>
600 DATA 67,82,79,76        <021>
602 DATA 76,33,76,83        <014>
604 DATA 67.82.79.76        <025>
606 DATA 76,33,82,83        <015>
608 DATA 67,82,79,76        <029>
610 DATA 76,33,83,69        <024>
612 DATA 84,33,82,69        <024>
614 DATA 83,69,84,33        <027>
616 DATA 68,82,65,87        <035>
618 DATA 33,67,68,82        <030>
620 DATA 65,87,33,67        <034>
622 DATA 76,69,65,82        <040>
624 DATA 162,5,169          <187>
626 DATA 0,157,82,3         <229>
628 DATA 202,16,250         <227>
630 DATA 162,23,14          <182>
632 DATA 79,3,46,80         <246>
634 DATA 3,46,81,3          <188>
636 DATA 46,82,3,46         <246>
638 DATA 83,3,46,84         <251>
640 DATA 3,14,85,3          <193>
642 DATA 46,86,3,46         <000>
644 DATA 87,3,56,173        <053>
646 DATA 82,3,237,88        <056>
648 DATA 3,168,173          <210>
650 DATA 83,3,233,0         <249>
652 DATA 141,94,3,173       <102>
654 DATA 84,3,233,0         <254>
656 DATA 144,37,140         <005>
658 DATA 82,3,141,84        <058>
660 DATA 3,173,94,3         <011>
662 DATA 141,83,3,24        <057>
664 DATA 169,1,109          <224>
666 DATA 85,3,141,85        <070>
668 DATA 3,169,0,109        <066>
670 DATA 86,3,141,86        <076>
672 DATA 3,169,0,109        <070>
674 DATA 87,3,141,87        <082>
676 DATA 3,202,16,166       <120>
678 DATA 96,56,185          <251>
680 DATA 80,3,73,255        <082>
682 DATA 105,0,153          <230>
684 DATA 80,3,200,202       <118>
686 DATA 208,242,96         <044>
688 DATA 24,173,85          <251>
690 DATA 3,109,91,3         <037>
692 DATA 141,91,3,173       <139>
694 DATA 86,3,109,92        <101>
696 DATA 3,141,92,3         <040>
698 DATA 173,87,3,109       <154>
700 DATA 93,3,141,93        <102>
702 DATA 3,96,24,169        <111>
704 DATA 128,109,92         <061>
706 DATA 3,169,0,109        <104>
708 DATA 93,3,96,234        <116>
710 DATA 8,32,202,32        <101>
712 DATA 162,1,56,189       <168>
714 DATA 89,3,253,60        <119>
716 DATA 3,8,176,4          <018>
718 DATA 73,255,105         <071>
720 DATA 1,157,76,3         <071>
722 DATA 202,16,236         <069>
724 DATA 13,77,3,208        <124>
726 DATA 4,40,40,40         <063>
728 DATA 96,169,1,141       <183>
730 DATA 78,3,162,2         <080>
732 DATA 169,0,157          <038>
734 DATA 91,3,202,16        <127>
736 DATA 250,173,76         <092>
738 DATA 3,205,77,3         <086>
740 DATA 176,60,169         <101>
```

```
742 DATA 106,141,0               <032>
744 DATA 1,169,37,141            <194>
746 DATA 1,1,173,60              <086>
748 DATA 3,141,93,3              <093>
750 DATA 173,76,3,141            <200>
752 DATA 81,3,173,77             <158>
754 DATA 3,141,88,3              <103>
756 DATA 169,0,141               <055>
758 DATA 79,3,141,80             <160>
760 DATA 3,32,0,36               <050>
762 DATA 40,176,7,162            <213>
764 DATA 3,160,5,32              <105>
766 DATA 103,36,40               <060>
768 DATA 176,5,169               <081>
770 DATA 255,141,78              <129>
772 DATA 3,76,89,37              <137>
774 DATA 169,133,141             <177>
776 DATA 0,1,169,37              <125>
778 DATA 141,1,1,173             <167>
780 DATA 61,3,141,93             <178>
782 DATA 3,40,176,5              <130>
784 DATA 169,255,141             <192>
786 DATA 78,3,173,77             <199>
788 DATA 3,141,81,3              <131>
790 DATA 173,76,3,141            <241>
792 DATA 88,3,169,0              <149>
794 DATA 141,79,3,141            <243>
796 DATA 80,3,32,0               <086>
798 DATA 36,40,176               <103>
800 DATA 7,162,3,160             <164>
802 DATA 5,32,103,36             <195>
804 DATA 173,76,3,141            <255>
806 DATA 94,3,76,95              <171>
808 DATA 37,173,77               <121>
810 DATA 3,141,94,3              <157>
812 DATA 108,0,1,206             <200>
814 DATA 94,3,208,248            <016>
816 DATA 76,160,37               <124>
818 DATA 24,173,78               <128>
820 DATA 3,109,61,3              <165>
822 DATA 141,61,3,32             <213>
824 DATA 119,36,32               <127>
826 DATA 148,36,141              <180>
828 DATA 60,3,40,8               <123>
830 DATA 32,202,32               <122>
832 DATA 76,98,37,24             <248>
834 DATA 173,78,3,109            <035>
836 DATA 60,3,141,60             <227>
838 DATA 3,32,119,36             <236>
840 DATA 32,148,36               <145>
842 DATA 141,61,3,40             <232>
844 DATA 8,32,202,32             <236>
846 DATA 76,98,37,173            <059>
848 DATA 89,3,141,60             <250>
850 DATA 3,173,90,3              <198>
852 DATA 141,61,3,40             <242>
854 DATA 32,202,32               <146>
856 DATA 96,32,117               <163>
858 DATA 233,76,234              <214>
860 DATA 199,234,234             <015>
862 DATA 234,0,32,104            <043>
864 DATA 32,192,33               <165>
866 DATA 177,37,80               <177>
868 DATA 34,160,34               <167>
870 DATA 2,38,18,38              <225>
872 DATA 34,38,50,38             <020>
874 DATA 40,32,32,121            <054>
876 DATA 0,201,40,240            <051>
878 DATA 7,201,44,240            <064>
880 DATA 3,76,39,35              <238>
882 DATA 32,155,215              <232>
884 DATA 201,41,208              <228>
886 DATA 3,32,115,0              <223>
888 DATA 138,96,32               <198>
890 DATA 208,37,72               <197>
892 DATA 201,168,176             <042>
894 DATA 10,32,208               <188>
896 DATA 37,201,176              <249>
898 DATA 176,3,168               <208>
900 DATA 104,96,104              <251>
902 DATA 162,14,76               <207>
904 DATA 55,196,32               <213>
906 DATA 234,37,141              <001>
908 DATA 60,3,140,61             <043>
910 DATA 3,56,32,202             <047>
912 DATA 32,76,174               <220>
914 DATA 199,32,234              <017>
916 DATA 37,141,60               <216>
918 DATA 3,140,61,3              <002>
920 DATA 24,32,202               <213>
922 DATA 32,76,174               <230>
924 DATA 199,32,234              <027>
926 DATA 37,141,89               <237>
928 DATA 3,140,90,3              <014>
930 DATA 56,32,161               <232>
932 DATA 36,76,174               <244>
934 DATA 199,32,234              <037>
936 DATA 37,141,89               <247>
938 DATA 3,140,90,3              <024>
940 DATA 24,32,161               <237>
942 DATA 36,76,174               <254>
944 DATA 199,32,91               <000>
946 DATA 228,76,229              <054>
948 DATA 34,234,234              <043>
950 DATA 40,67,41,66             <098>
952 DATA 89,32,74,46             <109>
954 DATA 83,39,56,52             <109>
956 DATA 234,234,234             <101>
1000 FOR I=8192 TO 9816:READ DC
   :POKE I,DC:S=S+DC:NEXT I      <148>
1010 IF S<>159413 THEN PRINT"DATA
   [SPACE]FEHLER":END            <124>
1040 PRINT"BEREIT[SPACE]ZUM[SPACE]
   ABSAVEN[SPACE]?"              <231>
1045 GET A$:IF A$<>"J" THEN 1045 <172>
1050 POKE 43,255:POKE 44,31:POKE 45,90
   :POKE 46,38:SAVE"MINI[SPACE]
   GBASIC",1,1                   <240>
```

Delete

Diese in Maschinensprache geschriebene Routine ermöglicht es, Basic-Programmzeilen in einem vorzugebenden Zeilennummernbereich zu löschen.

Ein ärgerlicher Nachteil des C 64-Basic und auch von Simons-Basic ist das Fehlen einer Delete-Routine zum schnellen Löschen mehrerer Programmzeilen. Das folgende Programm hilft dem ab.

Der Basic-Lader speichert das Maschinenprogramm im Kassettenpuffer ab Adresse 828 (dez.). Das Programm kann dann auf drei verschiedene Arten aufgerufen werden:

— SYS 828, ZN - ZN Bereich löschen
— SYS 828, - ZN Bis Zeile alles löschen
— SYS 828, ZN - Ab Zeile alles löschen

Es werden jeweils die Zeilen inklusive der angegebenen gelöscht. Da die Routine im Kassettenpuffer untergebracht ist, belegt sie keinen Basic-Speicherplatz.

(Hans-Herbert Hagedorn / ev)

```
10 REM    ***********************         <053>
15 REM    *                     *         <242>
20 REM    *       DELETE        *         <170>
25 REM    *                     *         <252>
30 REM    *  H.H.HAGEDORN       *         <053>
35 REM    *                     *         <006>
40 REM    *  RUPPRECHTSTR.30    *         <082>
45 REM    *                     *         <016>
50 REM    *  83 LANDSHUT        *         <227>
55 REM    *                     *         <026>
60 REM    *  TEL. 0871/67337    *         <059>
65 REM    *                     *         <036>
70 REM    ***********************         <113>
75 :                                      <133>
80 FOR I=828 TO 990 : READ A : POKE I,A   <061>
85 S=S+A : NEXT                           <084>
90 IF S <> 17132 THEN PRINT "DATENFEHLER" : END
                  <250>
95 PRINT "OK"                             <214>
100 DATA 032,253,174,032,121,000,144,006  <206>
105 DATA 240,004,201,171,208,023,032,107  <210>
110 DATA 169,032,019,166,165,095,133,025  <249>
115 DATA 165,096,133,026,032,121,000,240  <227>
120 DATA 004,201,171,240,005,162,011,076  <226>
125 DATA 058,164,032,115,000,032,107,169  <245>
130 DATA 208,243,165,020,005,021,208,008  <244>
135 DATA 169,255,133,020,133,021,208,006  <253>
140 DATA 230,020,208,002,230,021,032,019  <237>
145 DATA 166,165,095,133,036,165,096,133  <033>
150 DATA 037,056,165,036,229,025,165,037  <033>
155 DATA 229,026,144,201,165,045,229,036  <030>
160 DATA 133,095,165,046,229,037,133,096  <047>
165 DATA 024,165,025,101,095,133,045,165  <033>
170 DATA 026,101,096,133,046,160,000,177  <033>
175 DATA 036,145,025,230,025,208,002,230  <029>
180 DATA 026,230,036,208,002,230,037,056  <039>
185 DATA 165,095,233,001,133,095,165,096  <067>
190 DATA 233,000,133,096,016,225,032,089  <057>
195 DATA 166,032,051,165,076,145,227,000  <064>
200 DATA 000,000,000                      <083>
```

Commodore-Basic erweitert

Mit dem hier vorgestellten Maschinenprogramm wird der Basic-Befehlssatz des VC 20 oder des C 64 um sechs Befehle erweitert.

Die 6 Befehle lauten, in die Basic-Schreibweise übersetzt, GOTO N, GOSUB N, RESTORE N, READ D,A, READ N,D,A, und POP. Einen kleinen Nachteil muß man dabei allerdings in Kauf nehmen, denn diese Routinen kann man nur dem SYS-Befehl ansprechen. Es ist also nicht möglich, eine der Routinen direkt mit einem Basic-Befehlswort aufzurufen.

Der Zugriff auf diese Befehle kann insbesondere dann von großem Nutzen sein, wenn man Programme von anderen Computern umschreiben will, die diese Befehle benutzen.

Will man einen der neuen Befehle in einem Basic-Programm benutzen, muß man nur das Basic-Wort in der oben aufgeführten Liste durch ein »SYS (Adresse)« ersetzen. Die Parameter hinter dem Befehl werden genauso hinter den SYS-Befehl geschrieben, als ob sie hinter dem Basic-Befehl stehen würden.

Bei den nun folgenden Erläuterungen wird davon ausgegangen, daß sich das Maschinenprogramm im Kassettenpuffer ab Adresse 828 befindet. Wurde eine andere Anfangsadresse gewählt, ändern sich auch die Adressen der einzelnen Routinen.

Der Befehl GOTO N sieht in der Form, wie er im Programm verwendet werden muß, so aus: SYS(828)N; also doch noch recht einfach. Dieser Befehl bewirkt, daß man direkt zu einer beliebigen Zeile springen kann, deren Zeilennummer »N« vorher berechnet wurde. Nun zur Syntax. Bei diesem, wie auch bei allen folgenden Befehlen ist darauf zu achten, daß die Startadresse der Routine nach dem SYS-Befehl, (hier 828) in Klammern steht, um Sie von der darauf folgenden Parameterliste zu trennen und so als Adresse kenntlich zu machen. »N« repräsentiert hier, wie auch bei den weiter folgenden Befehlen, eine beliebige gültige numerische Variable, eine Zahl oder einen numerischen Ausdruck. Für »N« ist also beispielsweise auch der Ausdruck »INT(RND(1)*20)*10+100« erlaubt. Der Ausdruck muß nur einen Ganzzahlenwert zum Ergebnis haben. Noch zu bemerken ist, daß zwischen der geschlossenen Klammer der Adresse und der ersten Variablen oder dem Ausdruck kein Komma stehen darf. Das Komma wirkt wie bei PRINT oder READ wie ein Trennzeichen. Da dieser Befehl aber nur eine Variable oder einen Ausdruck enthalten darf, würde das zu einem »SYNTAX ERROR« führen. Dies gilt auch bei allen folgenden Befehlen. Ist die berechnete Zeilennummer nicht im Programm enthalten, erfolgt die Fehlermeldung »UNDEF'D STATEMENT ERROR«.

Für den Befehl GOSUB N gilt das gleiche, was auch zu GOTO N gesagt wurde, unter Berücksichtigung der Tatsache, daß es sich hier um einen Unterprogramm-Aufruf handelt. Mit diesem Befehl kann man also zu einer vorher berechneten Unterprogramm-Adresse springen (SYS(834) N).

RESTORE N ermöglicht es, den DATA-Zeiger auf eine bestimmte Zeile zu setzen. SYS(866)100 beispielsweise setzt den DATA-Zeiger auf das erste Datum der Zeile 100. Mit einem anschließenden READ-Befehl kann man dann gezielt auf die-

sen Datensatz zugreifen. Ist die angegebene Zeilennummer im Programm nicht vorhanden, erfolgt ein »UNDEF'D STATEMENT ERROR«.

READ D, A (SYS(890) d, a) liest direkt den D-ten DATA-Wert in die Variable A. Anstelle von »A« kann sowohl eine numerische als auch eine Stringvariable stehen. »SYS(890) 5, A$« entspricht beispielsweise der Basic-Befehlsfolge »FOR I = 1 TO 5 : READ A$: NEXT«. Auf etwas ist noch zu achten: Will man numerische Daten mit einer numerischen Variable lesen, darf keiner der vorhergehenden DATA-Werte ein String sein. Dies ist programmtechnisch bedingt und liegt daran, daß in Wirklichkeit die entsprechende Anzahl von READ-Befehlen durchgeführt wird. Ein direktes Lesen nur des gesuchten Datums würde das Maschinen-Programm dreimal so lang machen. Sollte es doch einmal vorkommen, daß man versucht, in eine numerische Variable einen String einzulesen, wird ein »SYNTAX ERROR« mit Angabe der entsprechenden DATA-Zeile ausgegeben. Am besten benutzt man immer Stringvariable zum Lesen.

Dann wäre da noch der Befehl »READ N,D,A« (entspricht SYS(927)N,D,A). Dieser Befehl ist eine Mischung des RESTORE- und des READ-Befehls.

SYS(927)N,D,A liest aus der Zeile N das D-te Datum dieser Zeile in die Variable »A«. Ist D größer als die Anzahl der Daten in dieser Zeile, wird in der nächsten DATA-Zeile weitergelesen.

Nun noch zu »POP« (SYS(937)). Springt man aus einem Unterprogramm anstatt mit »RETURN« mit einem direkten Sprungbefehl in die nächsthöhere Ebene (zum Beispiel ins Hauptprogramm) zurück, dann kann mit SYS(937) die letzte gespeicherte Rücksprungadresse im Stack, die dann nicht mehr gebraucht wird, gelöscht werden. Damit wird verhindert, daß der Stack überläuft, da er maximal 23 Rücksprungadressen speichern kann. Außerdem wird ein korrekter Programmablauf sichergestellt, wenn man »hart« aus einem Unterprogramm herausspringt, beispielsweise, um in eine Fehlerbehandlungsroutine zu gehen.

Tips für die Eingabe

Als erstes sollte man nur den Basic-Lader (Listing 1 oder 2, je nach Computer) ab Zeile 10 000 eintippen und dann, ohne ihn zu starten, sicherheitshalber erst mal abspeichern. Dann gibt man noch eine Testzeile ein, und zwar: »10 GOSUB 10 000:PRINT"Prüfsumme=";AS:END« und startet das Ganze mit »RUN«. Ergibt sich für die Prüfsumme ein anderer Wert als 18 413 für den VC 20 oder 17 901 für den C64, dann hat man sich irgendwo vertippt und die Datazeilen sind mit dem abgedruckten Listing noch einmal zu vergleichen. Eine genaue Kontrolle sollte man sowieso vornehmen, da sich durch Zufall eine richtige Prüfsumme ergeben kann, obwohl vielleicht zwei Werte falsch sind, die sich aber gegeneinander aufheben.

Sind alle Datazeilen fehlerfrei, löscht man Zeile 10 und speichert das Programm noch einmal ab, damit man den Basic-Lader an jedes gewünschte Programm anhängen kann.

Ist dies geschehen, kann man das Umrechnungsprogramm (Listing 3) eingeben. Es dient dazu, das Maschinenprogramm aus dem Kassettenpuffer an eine andere Stelle im Speicher zu verschieben. Der Kassettenpuffer hat ja den Nachteil, daß das Maschinenprogramm bei jeder Kassettenoperation zerstört wird. Außerdem können auch andere Befehlserweiterungen diesen Bereich benutzen, um Werte zwischenzuspeichern. Dadurch würde dann das Maschinenprogramm auch zerstört. Werden in einem Programm, das den Basic-Lader enthält, noch weitere DATA-Zeilen verwendet, so ist sicherzustellen, daß deren Zeilennummern größer sind als die höchste Zeilennummer des Basic-Laders, da sonst falsche Werte eingelesen würden.

Und noch ein Tip. Um die Adressen der Befehle nicht ändern zu müssen, wenn man das Maschinenprogramm in einen anderen Speicherbereich verlegt, verwendet man am besten Variablen, denen man am Anfang des Programms die Adresse zuweist. Dies könnte zum Beispiel so aussehen: »SM%=828«.Für die einzelnen Befehle würde dann folgendes gelten:

GOTO N	=	SYS(SM%)
GOSUB N	=	SYS(SM%+6)
RESTORE N	=	SYS(SM%+38)
READ D,A	=	SYS(SM%+62)
READ N,D,A	=	SYS(SM%+99)
POP	=	SYS(SM%+109)

Da alle diese Routinen weitgehend in das Betriebssystem des Computers eingebunden sind, werden bei Fehlern in der Ausführung die entsprechenden Systemfehlermeldungen ausgegeben.

(Wolfgang Thauer / ev)

```
1 REM BASIC-LADER VC 20                          <130>
2 REM                                            <145>
3 REM WOLFGANG THAUER                            <176>
4 REM AM SCHIEDSBERG 45                          <173>
5 REM 5205 ST.AUGUSTIN 2                         <215>
6 REM                                            <149>
7 REM                                            <150>
10000 AS=0:FOR I=828+AB TO(828+144)+AB:READ AP
   :POKE I,AP:AS=AS+AP:NEXT I:RETURN             <126>
10005 DATA 32                                    <036>
10010 DATA 198,3:REM**                           <226>
10020 DATA 76,163,200,169,3,32,251,195,165,123,
   72,165,122,72,165,58,72                       <061>
10030 DATA 165,57,72,169,141,72,32,121,0,32
                  <002>
10040 DATA 198,3:REM**                           <000>
10050 DATA 32,163,200,76,174,199,32              <150>
10060 DATA 198,3:REM**                           <020>
10070 DATA 32,19,198,176,3,76,227,200,165,95,
   233,1,133,65,165,96,233,0,133,66,96,32  <247>
10080 DATA 29,200,32                             <196>
10090 DATA 198,3:REM**                           <050>
10100 DATA 165,20,133,253,166;21,232,134,254,32,
   253,206,32,6,204                              <071>
10110 DATA 198,253,240,11,32,28,204,198,253,208,
   249,198,254,208,245,96,32                     <060>
10120 DATA 98,3 :REM**                           <031>
10130 DATA 32,253,206,32                         <188>
10140 DATA 125,3:REM**                           <090>
10150 DATA 96,169,255,133,74,32                  <062>
10170 DATA 192,3:REM**                           <124>
10180 DATA 154,201,141,240,3,76,224,200,104
                  <134>
10190 DATA 104,104,104,104,76,248,200,186,232,
   232,76,139,195,32,138,205,32,247,215,96 <196>
```

Listing 1. Basic-Lader »6 neue Befehle« (VC 20-Version)

```
1 REM BASIC LADER C 64                           <007>
2 REM                                            <145>
3 REM WOLFGANG THAUER                            <176>
4 REM AM SCHIEDSBERG 45                          <173>
5 REM 5205 ST.AUGUSTIN 2                         <215>
6 REM                                            <149>
7 REM                                            <150>
10000 AS=0:FOR I=828 TO(828+144)+AB:READ AP
   :POKE I,AP:AS=AS+AP:NEXT I:RETURN             <081>
10005 DATA 32                                    <036>
10010 DATA 198,3:REM**                           <226>
10020 DATA 76,163,168,169,3,32,251,163,165,123,
   72,165,122,72,165,58,72                       <069>
10030 DATA 165,57,72,169,141,72,32,121,0,32
                  <002>
10040 DATA 198,3:REM**                           <000>
10050 DATA 32,163,168,76,174,167,32              <158>
10060 DATA 198,3:REM**                           <020>
10070 DATA 32,19,166,176,3,76,227,168,165,95,
   233,1,133,65,165,96,233,0,133,66,96,32  <255>
10080 DATA 29,168,32                             <209>
10090 DATA 198,3:REM**                           <050>
10100 DATA 165,20,133,253,166,21,232,134,254,32,
```

```
   253,174,32,6,172                                  <079>
10110 DATA 198,253,240,11,32,28,172,198,253,208,
   249,198,254,208,245,96,32                         <064>
10120 DATA 98,3:REM**                                <031>
10130 DATA 32,253,174,32                             <192>
10140 DATA 125,3:REM**                               <090>
10150 DATA 96,169,255,133,74,32                      <062>
10160 DATA 192,3:REM**                               <114>
10170 DATA 154,201,141,240,3,76,224,168,104
             <137>
10180 DATA 104,104,104,104,76,248,168,186,232,
   232,76,139,163,32,138,173,32,247,183,96 <202>
```

Listing 2. Basic-Lader »6 neue Befehle« (C 64-Version)

```
1 REM ADRESSEN UMRECHNEN                             <138>
2 REM                                                <145>
3 REM WOLFGANG THAUER                                <176>
4 REM AM SCHIEDSBERG 45                              <173>
5 REM 5205 ST.AUGUSTIN 2                             <215>
6 REM                                                <149>
10 :                                                 <068>
20 REM DAS MACHINENPROGRAMM IST 145 BYTES LANG
                         <086>
30 :                                                 <088>
40 REM ZEICHERERKLAERUNG                             <177>
50 REM CHR$(147)= CLEAR HOME                         <124>
60 REM CHR$(17) = CURSOR DOWN                        <216>
70 REM CHR$(18) = REVERSE ON                         <134>
80 REM CHR$(13)  = RETURN                            <178>
90 :                                                 <148>
100 :                                                <158>
110 :                                                <168>
120 PRINT CHR$(147)                                  <197>
130 PRINT"DIESES[SPACE]PROGRAMM[SPACE]RECHNET
   [SPACE]DIE[SPACE]ABSOLUTEN[SPACE]ADRESSEN
   [SPACE]IN[SPACE]DER[SPACE]MASCHINENROU";
                         <151>
135 PRINT"TINE[SPACE]FUER[SPACE]EINEN[SPACE]
   ANDEREN[SPACE]SPEICHERBEREICH[SPACE]UM[SPACE]
   UND[SPACE]POKET[SPACE]ES";                        <086>
140 PRINT"LAUFFAEHIG[SPACE]IN[SPACE]DIESEN
   [SPACE]SPEICHERBEREICH."                          <247>
150 PRINT CHR$(17);"GEBE[SPACE]DIE[SPACE]
   GEWUENSCHTE[SPACE2]ANFANGSADRESSE[SPACE]EIN,
   AB"                                               <229>
160 PRINT"DER[SPACE]DAS[SPACE]MASCHINENPRO-
   [SPACE]GRAMM[SPACE]IM[SPACE]SPEICHER[SPACE5]
   LIEGEN[SPACE]SOLL."                               <093>
170 PRINT CHR$(17);"(DIE[SPACE]NORMALE[SPACE]
   ADRESSE[SPACE3]IST[SPACE]828[SPACE]=[SPACE]
   KASSETTEN-[SPACE2]PUFFER)."                       <030>
180 PRINT CHR$(17):INPUT AA                          <014>
190 PRINT CHR$(17);"DIE[SPACE]ANFANGSADRESSE
   [SPACE4]IST[SPACE]";AA                            <218>
200 PRINT CHR$(17);"IST[SPACE]DAS[SPACE]RICHTIG
   [SPACE]?"                                         <113>
210 INPUT"(J/N)";A$                                  <083>
220 IF A$<>"J"AND A$<>"N"THEN 210                    <002>
230 IF A$="N"THEN PRINT:GOTO 150                     <179>
240 AB=AA-828                                        <244>
245 PRINT CHR$(147);"BITTE[SPACE]WARTEN"             <010>
250 GOSUB 700                                        <030>
260 IF AA=828 THEN 490                               <170>
270 :                                                <073>
280 REM ANGABE DER ZU AENDERNDEN DATAZEILEN
                         <133>
290 REM DIESE DATAZEILEN SIND IM LISTING DURCH
   EIN ANGEHAENGTES    ':REM**'GEKENNZEICHNET
                         <105>
300 :                                                <103>
310 PRINT CHR$(17);"DIE[SPACE]ZU[SPACE]
   AENDERNDEN[SPACE]DATAZEILEN[SPACE]MIT[SPACE]
   DEN[SPACE]ENT-[SPACE3]SPRECHENDEN[SPACE]
   NEUEN"                                            <132>
320 PRINT"WERTEN[SPACE]LAUTEN:"                      <246>
330 PRINT                                            <228>
340 AD=AA+138:GOSUB 630                              <180>
350 PRINT"10010[SPACE]DATA";AL;",";AH                <127>
360 PRINT"10040[SPACE]DATA";AL;",";AH                <140>
370 PRINT"10060[SPACE]DATA";AL;",";AH                <152>
380 PRINT"10090[SPACE]DATA";AL;",";AH                <165>
390 AD=AA+38:GOSUB 630                               <181>
400 PRINT"10120[SPACE]DATA";AL;",";AH                <179>
410 AD=AA+65:GOSUB 630                               <201>
420 PRINT"10140[SPACE]DATA";AL;",";AH                <201>
430 AD=AA+132:GOSUB 630                              <008>
440 PRINT"10170[SPACE]DATA";AL;",";AH                <224>
450 PRINT CHR$(17);"WEITER[SPACE]MIT[SPACE]
   <RETURN>"                                         <111>
460 GET A$:IF A$=""THEN 460                          <052>
470 IF A$<>CHR$(13)THEN 460                          <232>
480 :                                                <027>
490 PRINT                                            <132>
500 PRINT"DIE[SPACE]STARTADRESSEN[SPACE]DER
   [SPACE]ROUTINEN[SPACE]LAUTEN:"                    <217>
510 PRINT CHR$(17);"GOTO[SPACE]N[SPACE2]=[SPACE]
   SYS(";AA;")";CHR$(17)                             <162>
520 PRINT"GOSUB[SPACE]N[SPACE]=[SPACE]SYS(";
   AA+6;")";CHR$(17)                                 <025>
530 PRINT"RESTORE[SPACE]N=SYS(";AA+38;")";
   CHR$(17)                                          <252>
540 PRINT"RESTORE[SPACE]D,A$"
   :PRINT"[SPACE8]=[SPACE]SYS(";AA+62;")";
   CHR$(17)                                          <161>
550 PRINT"RESTORE[SPACE]N,D,A$"
   :PRINT"[SPACE8]=[SPACE]SYS(";AA+99;")";
   CHR$(17)                                          <047>
560 PRINT"POP[SPACE5]=[SPACE]SYS(";AA+109;")"
                         <011>
570 END                                              <188>
580 :                                                <128>
590 :                                                <138>
600 REM SUBROUTINE                                   <249>
610 REM BERECHNUNG VON LOW- UND HIGH-BYTE EINER
   ADRESSE                                           <200>
620 :                                                <168>
630 AH=INT(AD/256):AL=AD-AH*256:RETURN               <203>
640 :                                                <188>
650 :                                                <198>
670 REM SUBROUTINE                                   <063>
680 REM MASCHINENPROGRAMM WIRD IN SPEICHER GEPO
   KET                                               <099>
690 :                                                <238>
700 GOSUB 10000                                      <060>
710 IF AS=18413 THEN PRINT CHR$(17);"PRUEFSUMME
   [SPACE]KORREKT.":RETURN                           <250>
712 REM  18413 = PRUEFSUMME FUER VC 20               <205>
714 REM FUER C 64 IST DIE PRUEFSUMME 17901           <007>
720 PRINT CHR$(147);"PRUEFSUMME[SPACE]=";AS
                         <179>
730 PRINT CHR$(17);"PRUEFSUMMENFEHLER[SPACE]!"
                         <162>
740 PRINT                                            <127>
750 PRINT"MOEGLICHE[SPACE]FEHLER-[SPACE5]
   QUELLEN[SPACE]:"                                  <141>
760 PRINT                                            <147>
770 PRINT"*EINE[SPACE]DATE[SPACE]WURDE[SPACE]
   VER-[SPACE2]GESSEN[SPACE]ODER[SPACE]FALSCH
   [SPACE4]EINGE-TIPPT."                             <224>
780 PRINT                                            <168>
790 PRINT"*SIE[SPACE]HABEN[SPACE]DIE[SPACE2]
   DATA-[SPACE2]ZEILEN[SPACE]GEAENDERT,[SPACE]
   UM[SPACE2]DAS[SPACE]MASCHINENPROGRAMM";           <170>
800 PRINT"[SPACE]AN[SPACE]EINEN[SPACE]ANDEREN
   [SPACE6]SPEICHERBEREICH[SPACE]ANZU-[SPACE]
   PASSEN."                                          <131>
810 PRINT"*SIE[SPACE]HABEN[SPACE]DIE[SPACE]
   FALSCHE[SPACE]PRUEFSUMME[SPACE]IN[SPACE]
   ZEILE[SPACE3]710[SPACE]EINGESETZT."               <028>
820 END                                              <183>
830 :                                                <123>
960 :                                                <253>
970 REM UNTERPROGRAMM                                <079>
980 REM BASICLADER                                   <048>
990 :                                                <027>

10000 AS=0:FOR I=828+AB TO(828+144)+AB:READ AP
   :POKE I,AP:AS=AS+AP:NEXT I:RETURN                 <126>
10005 DATA 32                                        <036>
10010 DATA 198,3:REM**                               <226>
10020 DATA 76,163,200,169,3,32,251,195,165,123,
   72,165,122,72,165,58,72                           <061>
10030 DATA 165,57,72,169,141,72,32,121,0,32
                         <002>
10040 DATA 198,3:REM**                               <000>
10050 DATA 32,163,200,76,174,199,32                  <150>
10060 DATA 198,3:REM**                               <020>
10070 DATA 32,19,198,176,3,76,227,200,165,95,
   233,1,133,65,165,96,233,0,133,66,96,32   <247>
10080 DATA 29,200,32                                 <196>
```

```
10090 DATA 198,3:REM**                         <050>
10100 DATA 165,20,133,253,166,21,232,134,254,32,
   253,206,32,6,204                            <071>
10110 DATA 198,253,240,11,32,28,204,198,253,208,
   249,198,254,208,245,96,32                   <060>
10120 DATA 98,3 :REM**                         <031>
10130 DATA 32,253,206,32                       <188>
10140 DATA 125,3:REM**                         <090>
10150 DATA 96,169,255,133,74,32                <062>
10170 DATA 192,3:REM**                         <124>
10180 DATA 154,201,141,240,3,76,224,200,104
                         <134>
10190 DATA 104,104,104,104,76,248,200,186,232,
   232,76,139,195,32,138,205,32,247,215,96 <196>
```

Listing 3. Der Adressen-Umrechner

Hardcopy im Superformat

Hier ist endlich eine formatfüllende Hardcopy-Routine für den Commodore 64 mit Epson-Drucker und Simons Basic

Das Programm erzeugt eine vergrößerte Hardcopy vom Grafikbildschirm. Die in Simons Basic enthaltene Hardcopy nutzt ja leider nur das halbe Blatt aus, was in vielen Fällen sehr störend ist. Das Programm vergrößert den Ausdruck in X-und Y-Richtung, so daß eine halbe A4-Seite bedruckt wird.

Es ist für den Commodore 64 und Epson-Drucker mit VC-Interface gedacht, läßt sich aber auch an andere Nadeldrucker anpassen.

(Peter Schwabe / ev)

Ein Hardcopy-Beispiel (Originalgröße)

Commodore 64

Auch selbsterstellte Grafiken sind schnell ausgedruckt

```
11 REM ***********************            <054>
12 REM ****                  ***          <193>
13 REM ****    HARDCOPY      ***          <028>
14 REM ****                  ***          <195>
15 REM ***********************            <058>
20 REM **** PETER SCHWABE ***             <070>
30 REM ****  IBBENBUEREN  ***             <244>
40 REM ****  12.09.1984   ***             <219>
50 REM ***********************            <093>
60 PRINT"[CLEAR,DOWN4,SPACE2]HARDCOPY[SPACE]
   FUER[SPACE]SIMON'S[SPACE]BASIC         <188>
70 PRINT"[DOWN2,SPACE2]DIE[SPACE]COPY[SPACE]
   WIRD[SPACE]MIT[SPACE]SYS[SPACE]32500[SPACE]
   GESTARTET                              <252>
80 PRINT"[DOWN2,SPACE2]BITTE[SPACE]WARTEN,
   [SPACE]PROGRAMM[SPACE]WIRD[SPACE]GELADEN
                 <045>
90 FOR K=32500 TO 32699                   <131>
100 READ A:POKE K,A:S=S+A:NEXT            <238>
320 DATA 169,126,162,  4,160,  1, 32,186,255,
   169,  0, 32                            <075>
330 DATA 192,255,162,126, 32,201,255,169,  0,
   160,224,133                            <070>
340 DATA 253,132,254,162, 50,169,  1,133,252,
   160,  2,185                            <237>
350 DATA 177,127, 32,210,255,136, 16,247,160,
     5,185,171                            <052>
360 DATA 127, 32,210,255,136, 16,247,169, 40,
   133, 21,169                            <058>
370 DATA 128,133,151,169,  0,133, 20,160,  3,
   169, 52,120                            <208>
380 DATA 133,  1,177,253, 37,151,240, 10,165,
    20, 25,180                            <219>
390 DATA 127, 25,184,127,133, 20,136, 16,237,
   169, 55,133                            <093>
400 DATA   1, 88,165, 20, 32              <200>
440 DATA 210,255, 32,210,255, 70,151,144,209,
   165,253,105                            <225>
450 DATA   7,133,253,144,  2,230,254,198, 21,
   208,191,165                            <099>
460 DATA 252,240, 22, 56,165,253,233, 60,133,
   253,165,254                            <206>
470 DATA 233,  1,133,254,169,  0,133,252,202,
   208,152,240                            <149>
480 DATA  20, 56,165,253,233,  4,133,253,165,
   254,233,  0                            <071>
490 DATA 133,254,169,  1,133,252,202,208,130,
   169, 13, 32                            <176>
500 DATA 210,255, 32,204,255,169,126, 76,195,
   255,  2,128                            <208>
510 DATA   4, 42, 27, 13, 23, 51, 27,128, 32,
     8,  2, 64                            <213>
520 DATA  16,  4,  1,-1                   <059>
600 IF S <> 26274 THEN PRINT "FEHLER[SPACE]IN
   [SPACE]DATAS[SPACE]!!!":END            <252>
610 PRINT"[DOWN3,SPACE2]PROGRAMM[SPACE]IST
   [SPACE]BEREIT[SPACE]!!!"               <231>
620 POKE 56,126:REM VOR UEBERSCHREIBEN SCHUETZEN
                  <180>
```

Listing »Super-Hardcopy«

Zeichen-Editor

Bei vielen Anwendungen ist es sinnvoll mit zwei verschiedenen Zeichensätzen zu arbeiten. Dieses Programm ermöglicht Ihnen, einen eigenen Zeichensatz zu erstellen, ohne den Original-Zeichensatz zu zerstören.

Programmiert man ein Videospiel, ein Textverarbeitungsprogramm oder will man einfach nur die üblichen Bildschirmzeichen etwas interessanter gestalten, bleibt einem nichts anderes übrig, als den normalen Zeichensatz aus dem ROM herauszuholen, ins RAM zu kopieren und dann in diesem kopierten Zeichensatz »herumzuPOKEn«.

Diese Tätigkeit ist aber — ähnlich wie bei der Konstruktion von Sprites — immer wieder eine mühsame Rechnerei. Deshalb habe ich mit einen komfortablen Zeichen-Editor zusammengestellt, der jede Rechnerei abnimmt. Mit seiner Hilfe ist das Definieren eigener Grafik-Zeichen ein Kinderspiel.

Nach dem Eintippen (und Abspeichern!) läßt man das Programm mit RUN starten. Nach einer kurzen Wartezeit, während der Großbuchstaben-Zeichensatz aus dem ROM ins RAM kopiert wird (und zwar in die Speicherzellen 51200 bis 53248, der Bildschirm beginnt dann bei Adresse 50176), erscheint das Menü mit einem Zeichenfeld links unten. (Wen es interessiert: das Maschinenprogramm, das den Zeichensatz ins RAM kopiert, beginnt bei der Adresse 828, also dem Anfang des Kassettenpuffers).

In diesem Zeichenfeld kann man nun mit den Cursortasten herumwandern, Sternchen (*) malen und gegebenenfalls mit der Space-Taste wieder löschen. Ein Sternchen im Zeichenfeld bedeutet einfach, daß hier ein Bit gesetzt wird, das dann später zur Berechnung des Zeichens dient. Jedes Graphik-Zeichen besteht ja aus 8 Bytes — und genau diese 8 Bytes stellt das Zeichenfeld symbolisch dar.

Taste 1: Berechnung des selbsterstellten Zeichens

Sobald man sein Zeichen gemalt hat, muß es berechnet werden. Hierfür ist im Menü der Programmpunkt 1 vorgesehen. Drückt man diese Taste, so wird gefragt: »Welche Taste?« Man drückt nun die Taste, der man das soeben erstellte Zeichen zuordnen möchte, und genau das Zeichen, das zu der gedrückten Taste gehört, wird nun durch das im Zeichenfeld definierte Zeichen ersetzt.

Ein Beispiel: Füllt man das Zeichenfeld ganz mit Sternchen aus, läßt das Feld berechnen und drückt dann die Taste »A«, so wird überall auf dem Bildschirm dort, wo eben noch ein »A« gestanden hat, ein reverses Quadrat erscheinen. Nette Spielchen kann man zum Beispiel mit der Space-Taste machen: dann wird nämlich überall da, wo ein Space auf dem Bildschirm ist (und das sind ja üblicherweise eine ganze Menge), das eben definierte Zeichen gedruckt. Auf diese Weise kann man den Bildschirm etwas interessanter gestalten.

Taste 2:

Sobald das neue Zeichen berechnet und ausgedruckt ist, kann man wieder beliebig im Zeichenfeld herumhantieren. Will man jedoch ein ganz anderes Zeichen konstruieren, so drückt man einfach Taste 2, und flugs ist das Zeichenfeld wieder »sauber« — das gerade definierte Zeichen wird dabei natürlich nicht gelöscht. Zusätzlich zum reinen Konstruieren von Grafik-Zeichen gibt es jodoch auch noch andere Programmfunktionen:

Taste 3: ROM-Zeichen auslesen

Mit Hilfe dieses Programmpunktes kann man sich ein beliebiges Zeichen aus dem ROM-Zeichensatz herholen. Hierbei wird das gewünschte Zeichen in das Zeichenfeld hineingePOKEt und die Daten dieses Zeichens rechts daneben ausgegeben. Selbst wenn Sie also den Buchstaben »A« als ein reverses Quadrat definiert haben sollten, so erscheint — sofern Sie Taste 3 und danach »A« drücken — auf dem Zeichenfeld das gute, alte »A« wieder (denn im ROM-Zeichensatz bleibt natürlich alles beim alten).

Taste 4: Eigene Zeichen auslesen

Natürlich können Sie nicht nur Zeichen aus dem ROM holen, sondern auch aus dem kopierten Zeichensatz, in dem Sie bisher munter herummanipuliert haben. Sie haben zum Beispiel aus dem »O« ein Smiley-Gesicht gemacht und wollen er gerne in vergrößerte Form wiedersehen, um etwa Korrekturen oder ähnliches vorzunehmen — bitte sehr! Drücken Sie die Taste 4, dann ein »O« und Ihr eigenes Zeichen steht im Zeichenfeld — samt den dazugehörigen Daten rechts nebenan.

Taste 8: Restore

Nun kann es vorkommen, daß Sie genug haben von Ihren eigenen Zeichen. Eigentlich wollten Sie jetzt ganz gern wieder die alten, »normalen« Zeichen anstelle der vielen Smiley-Gesichter und reversen Quadrate sehen.

Wenn Sie Taste 8 drücken, wird einfach der ROM-Zeichensatz wieder ins RAM kopiert; und da dieser Teil in Maschinensprache geschrieben ist, geht das ziemlich schnell vonstatten.

Will man übrigens nur ein einziges Zeichen wieder in den Ursprungszustand versetzen, so geht man am besten so vor: Man holt sich das (alte) Zeichen aus dem ROM (mit Taste 3) und läßt es mit Taste 1 wieder in den neuen Zeichensatz hineinkopieren. Mit dieser Methode kann man zum Beispiel auch einen Art Geheimcode entwickeln: Man tauscht einfach die Zeichen im ROM untereinander aus (statt eines A ein B, statt eines U ein X und so weiter). Nach einigem Umdefinieren bekommt man einen ganz eigenartigen Buchstabensalat auf dem Bildschirm.

Taste 9: Bild zeichnen

Nehmen wir an, Sie haben aus den etwas eintönigen Grafikzeichen des C 64 interessantere Zeichen zusammengebastelt: Leitern, Mauerwerke, Treppen und ähnliches. Sie würden aber ganz gerne sehen, wie diese Zeichen im Zusammenhang wirken; wie es etwa aussieht, wenn ein Leiterstück unter dem anderen steht, daneben ein Mauerwerk und so weiter. Dazu drücken Sie Taste 9 und können jetzt mit Ihren selbsterstellten Zeichen den ganzen Bildschirm vollmalen. Wenn Sie wieder ins Menü zurückwollen, drücken Sie einfach »Cursor Home« (steht auch auf dem Bildschirm).

Ein Hinweis zu diesem Programmpunkt: Der Cursor ist beim Bildmalen aus Gründen der Programmiervereinfachung nicht immer sehr gut sichtbar, manchmal »legt« er sich sogar ab, wenn er schnell über den Bildschirm bewegt wird. Das sollte Sie aber nicht weiter stören, schließlich ist diese Programmfunktion nur als Hilfe gedacht für einen schnellen Überblick.

Tasten 6 und 7: Abspeichern und Laden

Wenn Sie eigene Zeichen definiert haben, möchten Sie diesen Zeichensatz vielleicht abspeichern, um ihn später noch einmal verwenden zu können. Drücken Sie einfach Taste 6 und geben einen Dateinamen ein — der kopierte Zeichensatz mit Ihren selbsterstellten Zeichen wird nun auf Diskette gespeichert (nur der Großbuchstaben-Zeichensatz). Er nimmt dabei genau 44 Blöcke auf der Diskette ein. Mit Taste 7 können Sie schließlich einen abgespeicherten Zeichensatz wieder laden (der aktuelle Zeichensatz wird dabei überschrieben).

Sollten Sie einmal versehentlich 6 oder 7 gedrückt haben, so kommen Sie wieder ins Menü zurück, wenn Sie »N« + »RETURN« eingeben!

Taste 5: Daten eingeben

Eine letzte Programmfunktion bietet die Eingabe von Daten, die ein Zeichen definieren. Dies ist eine Alternative zum Konstruieren eines Zeichens im Zeichenfeld. Zu beachten ist, daß nur Zahlen von 0 bis 255 eingegeben werden können (dies wird allerdings vom Programm sichergestellt; negative und Zahlen größer als 255 werden vom Programm nicht angenommen). Nach Eingabe des 8. Bytes erscheint dann die übliche Frage »Welche Taste!« und nach Eingabe dieser Taste das bekannte Sternchenbild auf dem Zeichenfeld.

Die Erklärung der Taste 0 (Programmende) erübrigt sich wohl. Zu erwähnen ist hierbei nur, daß der kopierte Zeichensatz auch nach Beendigung des Programms weiterhin zur Bildschirmgestaltung benutzt wird; auch der Bildschirm selbst sitzt weiterhin an der Adresse 50176 (und nicht wie üblich, bei 1024). Wollen Sie also wieder einen ganz gewöhnlichen Bildschirm mit ganz gewöhnlichen Grafik-Zeichen haben, so geht das am einfachsten nur durch Aus- und Anschalten des Computers (auch die Run/Stop-Restore-Taste wurde im Programm blockiert).

Eine kleine Bemerkung am Rande zum eigenen Programmieren: Sehr oft kommt es ja vor, daß man einen Satz von einem Programm aus mitten in dem Bildschirm hineinschreiben will. Üblicherweise geht man so vor, daß man die Cursor-Steuerzeichen in die Print-Zeile so oft einfügt, bis der gesuchte Platz gefunden ist. Für Programmlistings ist das etwas unübersichtlich. Es gibt aber noch eine andere Methode, die ich in diesem Programm benutzt habe (und zwar am Anfang fast jedes Programmpunktes): will man zum Beispiel einen Buchstaben in die 14. Zeile, 5. Spalte schreiben, so gibt man zunächst

POKE 214,13

ein, danach ein PRINT-Kommando und dann PRINTTAB(4)»X«.

In Adresse 214 steht nämlich die Zeilenposition des Cursors. Um diese richtig einzusetzen, muß man allerdings noch einen PRINT-Befehl nachschicken: Cursor in Zeile 13 + PRINT-Befehl ergibt dann Zeile 14!

(Volker Bühn / gk)

Programmablauf-Plan:

Zeile	Beschreibung
1-7	Programmname und Adresse des Autors
10-30	Einlesen und Aufruf der Maschinenroutine, die den Zeichensatz vom ROM ins RAM an die Adresse 51200 kopiert
40-140	Menü und Zeichenfeld werden gezeichnet
300-460	Abfrage der Tastatur: 1. Zeichnen im Zeichenfeld 2. Aufruf der einzelnen Programmfunktionen
500-520	Unterprogramm zum Printen des Zeichenfeldes
1000-1530	Die einzelnen Unterprogramme zur Steuerung des Cursors, der Space-Taste und des Sternchens, um im Zeichenfeld ein Zeichen zu erzeugen.
2000-8040	**Einzelne Programmfunktionen:**
2000-2120	Berechnung eines erstellten Zeichens; Zuordnung dieses neuen Zeichens in den Zeichensatz; Printen der Daten des Zeichens
3000-3150	Ausgabe eines Zeichens aus dem ROM-Zeichensatz; Printen der Daten dieses Zeichens
4000-4150	Ausgabe eines Zeichens aus dem kopierten Zeichensatz, Printen der Daten dieses Zeichens
5000-5050	Neuer Bildschirm zum Zeichen mit dem selbsterstellten Zeichensatz. Die Zeile 5050 dient zur Cursor-Steuerung und Ausdrucken eines Zeichens auf dem Bildschirm
6000-6120	Abspeichern eines selbsterstellten Zeichensatzes auf Diskette
7000-7120	Laden eines Zeichensatzes von Diskette
8000-8040	Eingabe von 8 Bytes zur Definition eines Zeichens. In 8030 wird sichergestellt, daß keine Zahl größer als 8 Bit ist
10000-10040	POKEn des Maschinenprogramms nach 828
10100-10150	DATA-Zeilen des Maschinenprogramms (in Hexcodes)

Wichtige Programmzeilen:

Erstellung eines eigenen Zeichens:

2050-2070	Hier werden die Bytes des selbsterstellten Zeichens aus dem Zeichenfeld gelesen, berechnet und in B(1) bis B(8) abgelegt.
2080	Die Bytes des Zeichens werden nun in die Stelle des Zeichensatzes gePOKEt, die der Benutzer durch die Tastenabfrage definiert hat.
	ROM-Zeichen:
3050	Verhindern von Interrupts
3060-3070	Im ROM-Zeichensatz wird das vom Benutzer festgelegte Zeichen in B(1) bis B(8) sowie C(1) bis C(8) abgelegt
3080	Zulassen von Interrupts
3090-3110	Die Bytes B(1) bis B(8) werden in das Zeichenfeld in Form von Sternchen (*) hinein gePOKEt

Eigenes Zeichen holen:

4060-4070	wie 3060-3070, nur daß im kopierten Zeichensatz gesucht wird
4080-4100	wie 3090-3110
	Bild zeichnen:
5050	In den Adressen 209,210 und 211 steht die aktuelle Cursorposition. Dies verhindert, daß der Cursur auf den Bildschirm wie ein normales Zeichen gePRINTet wird.

Liste der wichtigsten Variablen

Z	Position des anfänglichen Cursors im Zeichenfeld (ist immer 50776)
PU$, NO$	Strings, um Punkte beziehungsweise Leerstellen zu PRINTen
A	Variable, die die aktuelle Cursorposition im Zeichenfeld angibt
G$	Stringvariable zur Tastaturabfrage
B(1) bis B(8), C(1) bis C(8)	Arrays, die die 8 Bytes eines Zeichens enthalten
G	Bildschirmcode eines Zeichens
C	Cursorposition beim Bildzeichnen
N$	Name einer einzulesenden beziehungsweise abzuspeichernden Datei
A$	Hexcode aus den DATA-Zeilen
HN,LN	Hi-Nibble beziehungsweise Lo-Nibble von A$
I,J,K	Schleifenvariable

```
1 REM ************************             <128>
2 REM * ZEICHEN-EDITOR/C64                 <187>
3 REM * VON     VOLKER BUEHN  *            <030>
4 REM *         B2,14         *            <236>
5 REM *         6800 MANNHEIM *            <003>
6 REM * AM      6.7.1984      *            <022>
7 REM ************************             <134>
10 PRINT"[CLEAR]":PRINT:PRINT"[SPACE2]BITTE
   [SPACE]WARTEN!":RESTORE:POKE 657,128
   :POKE 792,134:POKE 793,234              <062>
20 GOSUB 10000:REM EINLESEN DER
               MASCHINENROUTINE            <069>
30 SYS 828     :REM AUFRUF DERSELBEN       <132>
40 POKE 53280,0:POKE 53281,0
   :PRINT"[CLEAR,SPACE8,RVSON]SELBSTERSTELLTE
   [SPACE]ZEICHEN[RVOFF]"                  <034>
50 PRINT:PRINT"1[SPACE]ZEICHEN[SPACE]BERECHNEN
   [SPACE]6[SPACE]ABSPEICHERN"             <253>
60 PRINT"2[SPACE]NEUES[SPACE]ZEICHEN[SPACE5]7
   [SPACE]LADEN"                           <108>
70 PRINT"3[SPACE]ROM-ZEICHEN[SPACE7]8[SPACE]
   RESTORE"                                <211>
80 PRINT"4[SPACE]ZEICHEN[SPACE]HOLEN[SPACE5]9
   [SPACE]BILD[SPACE]ZEICHNEN"             <133>
90 PRINT"5[SPACE]DATAS[SPACE]EINGEBEN[SPACE4]0
   [SPACE]PROGRAMMENDE"                    <199>
100 Z=50776                                <121>
110 PU$="........[SPACE31]"                <157>
120 NO$="[SPACE28]"                        <047>
130 GOSUB 510:IF FL=1 THEN RETURN          <020>
140 A=Z:POKE A,PEEK(A)OR 128               <182>
300 GET G$:IF G$=""THEN 300                <153>
310 IF G$="[UP]"THEN GOSUB 1020            <171>
320 IF G$="[DOWN]"THEN GOSUB 1120          <054>
330 IF G$="[RIGHT]"THEN GOSUB 1220         <077>
340 IF G$="[LEFT]"THEN GOSUB 1320          <216>
350 IF G$="[SPACE]"THEN GOSUB 1420         <070>
360 IF G$="*"THEN GOSUB 1520               <123>
370 IF G$="1"THEN POKE A,PEEK(A)AND 127
    :GOTO 2000                             <093>
380 IF G$="2"THEN RUN 100                  <093>
390 IF G$="3"THEN 3000                     <016>
400 IF G$="4"THEN 4000                     <028>
405 IF G$="5"THEN 8000                     <038>
410 IF G$="6"THEN 6000                     <042>
420 IF G$="7"THEN 7000                     <054>
430 IF G$="8"THEN 30                       <221>
440 IF G$="9"THEN 5000                     <074>
450 IF G$="0"THEN PRINT"[CLEAR]":END       <176>
460 GOTO 300                               <233>
500 REM FELD ZEICHNEN                      <243>
510 PRINT"[HOME]":FOR I=1 TO 14:PRINT:NEXT <110>
520 FOR I=1 TO 8:PRINT PU$:NEXT:RETURN     <179>
1000 :                                     <037>
1010 REM CURSOR NACH OBEN                  <160>
1020 IF PEEK(A-40)=32 THEN RETURN          <057>
1030 POKE A,PEEK(A)AND 127:A=A-40
     :POKE A,PEEK(A)OR 128:RETURN          <205>
1100 :                                     <138>
1110 REM CURSOR NACH UNTEN                 <107>
1120 IF PEEK(A+40)=32 THEN RETURN          <157>
1130 POKE A,PEEK(A)AND 127:A=A+40
     :POKE A,PEEK(A)OR 128:RETURN          <048>
1200 :                                     <238>

1210 REM CURSOR NACH RECHTS                <014>
1220 IF PEEK(A+1)=32 THEN RETURN           <206>
1230 POKE A,PEEK(A)AND 127:A=A+1
     :POKE A,PEEK(A)OR 128:RETURN          <097>
1300 :                                     <083>
1310 REM CURSOR NACH LINKS                 <043>
1320 IF PEEK(A-1)=32 THEN RETURN           <052>
1330 POKE A,PEEK(A)AND 127:A=A-1
     :POKE A,PEEK(A)OR 128:RETURN          <199>
1400 :                                     <183>
1410 REM SPACE-TASTE IM ZEICHENFELD        <231>
1420 IF PEEK(A+1)=32 THEN POKE A,174:RETURN
                         <113>
1430 POKE A,46:A=A+1:POKE A,PEEK(A)OR 128:RETURN
                         <247>
1500 :                                     <027>
1510 REM "*" IM ZEICHENFELD                <159>
1520 POKE A,42:IF PEEK(A+1)=32 THEN POKE A,
     PEEK(A)OR 128:RETURN                  <124>
1530 A=A+1:POKE A,PEEK(A)OR 128:RETURN     <179>
2000 REM ******************                <132>
2001 REM *ZEICHEN BERECHNEN*               <075>
2002 REM ******************                <134>
2010 POKE 214,14:PRINT:PRINT TAB(14)"WELCHE
     [SPACE]TASTE?[SPACE]";                <110>
2020 GET G$:IF G$=""THEN 2020              <136>
2030 PRINT G$:G=PEEK(Z+28)*8:K=0:A=Z       <127>
2040 PRINT TAB(10);NO$:PRINT TAB(10);NO$
     :PRINT"[UP4]"                         <024>
2050 FOR J=1 TO 8:FOR I=A+7 TO A STEP-1    <214>
2060 IF PEEK(I)=42 THEN B(J)=B(J)+2↑K      <251>
2070 K=K+1:NEXT I:A=A+40:K=0:NEXT J        <047>
2080 FOR I=51200+G TO 51207+G:K=K+1:POKE I,B(K)
     :NEXT                                 <062>
2090 PRINT TAB(10);NO$                     <244>
2100 PRINT TAB(15);B(1);B(2);B(3);B(4)     <009>
2110 PRINT TAB(15);B(5);B(6);B(7);B(8)     <035>
2120 POKE(Z+11),G/8:FOR I=1 TO 8:B(I)=0:NEXT
     :FL=0:GOTO 140                        <115>
3000 REM **************                    <158>
3001 REM * ROM-ZEICHEN*                    <200>
3002 REM **************                    <160>
3010 GOSUB 510                             <240>
3020 POKE 214,14:PRINT:PRINT TAB(14)"WELCHE
     [SPACE]TASTE?[SPACE]";                <100>
3030 GET G$:IF G$=""THEN 3030              <128>
3040 K=0:A=Z:PRINT G$:G=PEEK(Z+28)*8       <117>
3050 POKE 56334,PEEK(56334)AND 254
     :POKE 1,PEEK(1)AND 251                <216>
3060 FOR I=53248+G TO 53255+G              <011>
3070 K=K+1:B(K)=PEEK(I):C(K)=B(K):NEXT:K=0 <030>
3080 POKE 1,PEEK(1)OR 4:POKE 56334,
     PEEK(56334)OR 1                       <043>
3090 FOR J=1 TO 8:FOR I=7 TO 0 STEP-1      <238>
3100 IF B(J)>=2↑I THEN B(J)=B(J)-2↑I
     :POKE(A+7-I),42                       <205>
3110 NEXT I:A=A+40:NEXT J                  <127>
3120 PRINT"[UP]"TAB(10);NO$                <211>
3130 PRINT TAB(15);C(1);C(2);C(3);C(4)     <023>
3140 PRINT TAB(15);C(5);C(6);C(7);C(8)     <049>
3150 FOR I=1 TO 8:B(I)=0:NEXT:GOTO 140     <239>
4000 REM *************************         <088>
4001 REM * EIGENES ZEICHEN HOLEN *         <015>
4002 REM *************************         <090>
4010 GOSUB 510                             <220>
4020 POKE 214,14:PRINT:PRINT TAB(14)"WELCHE
     [SPACE]TASTE?[SPACE]";                <080>
4030 GET G$:IF G$=""THEN 4030              <109>
4040 PRINT G$:G=PEEK(Z+28)*8:K=0:A=Z       <097>
4050 IF FL=1 THEN 4080                     <084>
4060 FOR I=51200+G TO 51207+G              <228>
4070 K=K+1:B(K)=PEEK(I):C(K)=B(K):NEXT:K=0 <010>

4080 FOR J=1 TO 8:FOR I=7 TO 0 STEP-1      <207>
4090 IF C(J)>=2↑I THEN C(J)=C(J)-2↑I
     :POKE(A+7-I),42                       <177>
4100 NEXT I:A=A+40:NEXT J                  <097>
4110 PRINT"[UP]"TAB(10);NO$                <181>
4120 IF FL=1 THEN 2080                     <153>
4130 PRINT TAB(15);B(1);B(2);B(3);B(4)     <255>
4140 PRINT TAB(15);B(5);B(6);B(7);B(8)     <025>
4150 POKE(Z+11),G/8:FOR I=1 TO 8:B(I)=0:NEXT
     :GOTO 140                             <187>
5000 REM ****************                  <202>
5001 REM *  BILD MALEN  *                  <007>
5002 REM ****************                  <204>
5010 PRINT"[CLEAR]";:POKE 53280,0:POKE 53281,1
     :POKE 646,0:POKE 650,128              <176>
5020 PRINT"[HOME=ZURUECK]"                 <211>
5030 POKE 204,0:GET G$:IF G$=""THEN 5030   <029>
5040 IF G$="[HOME]"THEN POKE 204,1:POKE 646,6
     :POKE 650,0:GOTO 40                   <181>
5050 C=PEEK(209)+PEEK(210)*256+PEEK(211)
     :POKE C,PEEK(C)AND 127:PRINT G$;:GOTO 5030
                   <061>
6000 REM *************                     <056>
6001 REM * SPEICHERN *                     <012>
6002 REM *************                     <058>
6010 PRINT"[CLEAR]":PRINT:PRINT"[SPACE7]
     SPEICHERN[SPACE]DES[SPACE]ZEICHENSATZES"
                   <072>
6020 PRINT:PRINT"(INPUT[SPACE]'N'[SPACE],[SPACE]
     WENN[SPACE]ZURUECK)"                  <085>
6030 PRINT:PRINT:PRINT"[SPACE2]NAME[SPACE]DER
     [SPACE]DATEI?"                        <202>
6040 INPUT N$                              <166>
6050 IF N$="N"THEN 40                      <005>
6060 OPEN 2,8,2,N$+",S,W"                  <228>
6070 CLOSE 1:OPEN 1,8,15:INPUT#1,DA
     :IF DA<>0 THEN 6010                   <160>
6080 FOR I=51200 TO 53248                  <245>
```

```
6090 S%=PEEK(I)                                  <103>
6100 PRINT#2,S%                                  <089>
6110 NEXT I                                      <192>
6120 CLOSE 2:GOTO 40                             <248>
7000 REM *********                               <124>
7001 REM * LADEN *                               <187>
7002 REM *********                               <126>
7010 PRINT"[CLEAR]":PRINT:PRINT"[SPACE7]LADEN
   [SPACE]EINS[SPACE]ZEICHENSATZES"              <074>
7020 PRINT:PRINT"(INPUT[SPACE]'N',[SPACE]WENN
   [SPACE]ZURUECK)"                              <065>
7030 PRINT:PRINT:PRINT"[SPACE2]NAME[SPACE]DER
   [SPACE]DATEI?"                                <182>
7040 INPUT N$                                    <146>
7050 IF N$="N"THEN 40                            <241>
7060 OPEN 2,8,2,N$+",S,R"                        <203>
7070 CLOSE 1:OPEN 1,8,15:INPUT#1,OA
   :IF OA<>0 THEN 7010                           <141>
7080 FOR I=51200 TO 53248                        <225>
7090 INPUT#2,S%                                  <039>
7100 POKE I,S%                                   <091>
7110 NEXT I                                      <172>
7120 CLOSE 2:GOTO 40                             <228>
8000 REM ******************                      <226>
8001 REM * DATAS EINGEBEN *                      <237>
8002 REM ******************                      <228>
8010 PRINT"[CLEAR]":PRINT:PRINT"[SPACE]DATEN
   [SPACE]FUER[SPACE]EIN[SPACE]ZEICHEN[SPACE]
   EINGEBEN"                                     <128>
8020 FOR I=1 TO 8:PRINT I".[SPACE]BYTE"
   :INPUT B(I):C(I)=B(I)                         <254>
8030 IF B(I)>255 OR B(I)<0 THEN I=I-1            <103>
8040 NEXT:FL=1:GOSUB 40:GOTO 4010                <107>
10000 REM ******************************         <094>
10001 REM * MASCHINENROUTINE-EINGABE *           <239>
10002 REM ******************************         <096>
10010 FOR I=0 TO 58:READ A$                      <036>
10020 HN=(ASC(LEFT$(A$,1))-48)*16
   :IF HN>144 THEN HN=HN-112                     <105>
10030 LN=ASC(RIGHT$(A$,1))-48
   :IF LN>9 THEN LN=LN-7                         <099>
10040 HN=HN+LN:POKE(828+I),HN:NEXT:RETURN        <126>
10100 DATA 78,A9,31,85,01,A9,00,85,64,A9         <240>
10110 DATA D0,85,63,A9,C8,85,65,A2,10,A0         <016>
10120 DATA 00,B1,62,91,64,C8,D0,F9,E6,63         <031>
10130 DATA E6,65,CA,D0,F2,A9,37,85,01,58         <061>
10140 DATA A9,12,8D,18,D0,A9,94,8D,00,DD         <083>
10150 DATA A9,C4,8D,88,02,20,44,E5,60            <153>
```

Listing »Zeichen-Editor«

Super Line — 80 Zeichen für den C 64

Ein kleiner Basic-Lader realisiert, wofür man sonst viel Geld ausgeben muß: 80 Zeichen pro Zeile

Es werden 4 neue Befehle definiert, die das Darstellen von 80 Zeichen möglich machen. Und das, ohne daß man lange programmieren muß.

Beginnen wir mit dem Einfachsten: dem Eingeben. Dies dürfte keine Schwierigkeiten bereiten. Bevor man das Quellenprogramm jedoch startet, sollte man es abspeichern, da es sich, vorausgesetzt, man hat keinen Fehler gemacht, selbst löscht. Anschließend kann man es mit RUN laufen lassen und wenn die Prüfsumme stimmt, erscheint nach einigen Sekunden einfach READY. Die neuen Befehle sind nun definiert und können angewendet werden.

Diese Befehle lauten:

O (für »ON«): dieser Befehl bewirkt ein Einschalten des 80-Zeichen Modus. Dabei wird auf hochauflösende Grafik umgeschaltet.

C (für »CLEAR«): der 80-Zeichen Bildschirm wird gelöscht.

W x,y,a$ (für »WRITE«): der String A$ wird an Spalte x Zeile y geschrieben.
x geht von 0 bis 79,
y geht von 0 bis 24.

F (für »OFF«): Abschalten des 80-Zeichen Modus.

So bewirkt zum Beispiel das kurze Programm:

```
10 O
20 C
30 W 0,0 "64'ER DAS MAGAZIN FÜR COMPUTER-FANS"
```

daß der in Anführungsstrichen stehende Satz in die linke obere Ecke geschrieben wird. Anschließend rührt sich nichts mehr und man kann durch die 'blinde' Eingabe von 'F' wieder zum normalen Bildschirm zurückkehren.

A$ läßt sich auch durch einzelne Stringvariable ersetzen, die mit + verknüpft werden, oder man kann auch eine normale numerische Variable verwenden. Allerdings dürfen keine Variable mit dem Namen O,C,W oder F verwendet werden. So ist zum Beispiel O$,CG,WR$ oder ähnliches verboten.

Nun einige detaillierte Angaben zum Programm selbst. Das Maschinenprogramm liegt im Bereich von $ 9000 bis $ 928F. Wer im Besitz eines Monitors ist, kann es direkt abspeichern und von der Diskette mit ,8,1 laden. Gestartet wird es dann mit SYS 36864. Wer das nicht will, der lädt einfach den Basic-Lader, der das Maschinenprogramm in den Speicher 'POKEt'. Zwischen $ 9000 und $ 902D wird zunächst der Basic-Vektor umgesetzt, und der Anfang des Basic-Speichers hochgelegt. Zwischen $ 902E und $ 9044 beginnt nun die Befehlsdekodierung. Bei Erkennen eines Befehls wird verzweigt, ansonsten in die normale Interpreterroutine gesprungen. Bei dem Befehl 'O' wird zunächst in der Unterroutine von $ 9233 bis 928E der Zeichensatz aus dem verdeckten Bereich $ D000 in den offenen Bereich $ C000 übertragen. Der Bereich von $ 0400 bis $ 0800 wird mit dem Code für die Hintergrundfarbe gefüllt. Außerdem wird bei $ 9057 das Register # 648 umgesetzt, damit es auf dem Bildschirm kein farbliches Durcheinander gibt. Weiterhin wird selbstverständlich der hochauflösende Grafik-Modus eingeschalten. Die Routine für den Befehl 'C' liegt zwischen $ 9081 und $ 90A3. Der Bereich der Bit-Map wird einfach mit 00 gefüllt.

Der Befehl 'F' wird zwischen $ 906C und $ 907E ausgeführt. Das Register 648 wird zurückgesetzt, der hochauflösende Grafik-Modus ausgeschaltet und der normale Bildschirm gelöscht.

Der Befehl, dessen Routine am längsten ist, ist der Befehl 'W'. Er wird zwischen $ 90A6 und $ 9230 bearbeitet. Zunächst werden die beiden Koordinaten x und y geholt und aus ihnen die Adresse der Bit Map berechnet, an der das erste Byte gesetzt wird. Dies geschieht zwischen $ 90A6 und 9135. Dann werden die einzelnen Zeichen des zu schreibenden Satzes geholt und ihr Code wird so umgerechnet, daß er mit der Stelle übereinstimmt, an der das jeweilige Zeichen in dem nach $ C000 verschobenen Zeichen ROM steht. Anschließend durchläuft jedes der 8 Bytes, aus denen ein Zeichen definiert ist, die gleiche Prozedur. Das Byte wird geholt, jedes zweite Bit ausgefiltert, und die verbliebenen 4 Bits zusammengeschoben. Das Zeichen ist jetzt nur noch durch 4 x 8 Punkte definiert. Jetzt müssen die entstandenen Nibbles noch in die Bytes der Bit Map gebracht werden. Dies geschieht mit einer EXOR-Verknüpfung. Dabei steuert ein Flag, das in $ 9300 steht, ob das Nibble in die linke oder die rechte Hälfte des Bytes geschrieben wird.

(Andreas Zell / rg)

```
1 REM **********************************                        <250>
2 REM ****      SUPER LINE             ****                      <152>
3 REM ****     ANDREAS ZELL            ****                      <023>
4 REM ****   SEGEBERGERSTR. 27         ****                      <078>
5 REM ****   8500 NUERNBERG 90         ****                      <194>
6 REM ****   TEL 0911/31 47 90         ****                      <252>
7 REM **********************************                        <000>
12 FOR T=36864 TO 37518:READ A:POKE T,A:B=B+A:NEXT               <245>
13 IF B<>87028 THEN PRINT"FEHLER[SPACE]IN[SPACE]DEN[SPACE]DATA
   [SPACE]ZEILEN":END                                            <138>
14 SYS 36864:END                                                <113>
100 DATA  169, 46, 141, 8, 3, 169, 144, 141, 9, 3                <185>
110 DATA  169, 1, 133, 43, 133, 45, 133, 47, 133, 49             <082>
120 DATA  169, 68, 133, 44, 133, 46, 133, 48, 133, 50            <148>
130 DATA  169, 0, 141, 0, 68, 141, 1, 68, 141, 2                 <146>
140 DATA  68, 169, 143, 133, 56, 96, 32, 115, 0, 201             <113>
150 DATA  79, 240, 15, 201, 70, 240, 14, 201, 87, 240            <158>
160 DATA  13, 201, 67, 240, 12, 76, 231, 167, 76, 81             <131>
170 DATA  144, 76, 108, 144, 76, 166, 144, 76, 129, 144          <054>
180 DATA  234, 32, 51, 146, 234, 234, 234, 169, 64, 141          <045>
190 DATA  136, 2, 32, 68, 229, 169, 59, 141, 17, 208             <175>
200 DATA  169, 24, 141, 24, 208, 76, 228, 167, 169, 4            <238>
210 DATA  141, 136, 2, 169, 27, 141, 17, 208, 169, 21            <231>
220 DATA  141, 24, 208, 32, 68, 229, 76, 228, 167, 169           <056>
230 DATA  32, 133, 252, 169, 0, 133, 251, 162, 0, 169            <242>
240 DATA  0, 129, 251, 24, 165, 251, 105, 1, 133, 251            <243>
250 DATA  165, 252, 105, 0, 133, 252, 201, 64, 240, 3            <250>
260 DATA  76, 137, 144, 76, 228, 167, 32, 155, 183, 224          <144>
270 DATA  80, 144, 3, 76, 72, 178, 138, 74, 133, 251             <254>
280 DATA  138, 41, 1, 240, 11, 234, 234, 234, 234, 169           <086>
290 DATA  1, 141, 0, 147, 208, 5, 169, 0, 141, 0                 <042>
300 DATA  147, 32, 253, 174, 24, 32, 158, 183, 134, 252          <169>
310 DATA  224, 25, 176, 3, 76, 220, 144, 76, 72, 178             <038>
320 DATA  169, 0, 133, 253, 133, 254, 165, 252, 133, 253         <233>
330 DATA  162, 0, 6, 253, 38, 254, 232, 224, 5, 208              <247>
340 DATA  247, 162, 0, 142, 1, 149, 165, 252, 141, 0             <043>
350 DATA  149, 14, 0, 149, 46, 1, 149, 232, 224, 3               <218>
360 DATA  208, 245, 24, 165, 253, 109, 0, 149, 133, 253          <226>
370 DATA  165, 254, 109, 1, 149, 133, 254, 24, 165, 253          <240>
380 DATA  101, 251, 133, 251, 165, 254, 105, 0, 133, 252         <018>
390 DATA  24, 162, 0, 6, 251, 38, 252, 232, 224, 3               <249>
400 DATA  208, 247, 24, 165, 252, 105, 32, 133, 252, 32          <002>
410 DATA  253, 174, 32, 158, 173, 36, 13, 48, 6, 32              <082>
420 DATA  221, 189, 32, 135, 180, 32, 166, 182, 134, 253         <081>
430 DATA  132, 254, 133, 255, 160, 0, 177, 253, 201, 191         <080>
440 DATA  144, 6, 56, 233, 128, 76, 109, 145, 201, 63            <210>
450 DATA  144, 6, 56, 233, 64, 76, 109, 145, 234, 234            <225>
460 DATA  234, 234, 234, 234, 234, 153, 0, 148, 200, 198         <115>
470 DATA  255, 240, 3, 76, 80, 145, 169, 0, 153, 0               <083>
480 DATA  148, 234, 234, 234, 234, 234, 160, 0, 185, 0           <031>
490 DATA  148, 208, 8, 234, 234, 234, 234, 234, 76, 174          <111>
500 DATA  167, 133, 253, 169, 0, 133, 254, 6, 253, 38            <016>
510 DATA  254, 6, 253, 38, 254, 6, 253, 38, 254, 24              <187>
520 DATA  165, 254, 105, 192, 133, 254, 162, 0, 142, 1           <069>
530 DATA  147, 161, 253, 41, 1, 133, 255, 161, 253, 41           <081>
540 DATA  4, 74, 69, 255, 133, 255, 161, 253, 41, 16             <008>
550 DATA  74, 74, 69, 255, 133, 255, 161, 253, 41, 64            <076>
560 DATA  74, 74, 74, 69, 255, 133, 255, 173, 0, 147             <039>
570 DATA  201, 1, 240, 8, 6, 255, 6, 255, 6, 255                 <083>
```

```
580 DATA  6, 255, 165, 255, 65, 251, 129, 251, 238, 1           <100>
590 DATA  147, 24, 165, 251, 105, 1, 133, 251, 165, 252         <192>
600 DATA  105, 0, 133, 252, 24, 165, 253, 105, 1, 133           <089>
610 DATA  253, 165, 254, 105, 0, 133, 254, 173, 1, 147          <164>
620 DATA  201, 8, 240, 3, 76, 175, 145, 173, 0, 147             <028>
630 DATA  201, 1, 208, 7, 200, 142, 0, 147, 76, 132             <020>
640 DATA  145, 169, 1, 141, 0, 147, 56, 165, 251, 233           <149>
650 DATA  8, 133, 251, 165, 252, 233, 0, 133, 252, 200          <193>
660 DATA  76, 132, 145, 120, 169, 0, 133, 251, 133, 253         <005>
670 DATA  169, 208, 133, 252, 169, 192, 133, 254, 169, 51       <141>
680 DATA  133, 1, 162, 0, 161, 251, 129, 253, 24, 165           <178>
690 DATA  251, 105, 1, 133, 251, 165, 252, 105, 0, 133          <227>
700 DATA  252, 24, 165, 253, 105, 1, 133, 253, 165, 254         <049>
710 DATA  105, 0, 133, 254, 201, 200, 208, 222, 169, 55         <046>
720 DATA  133, 1, 88, 169, 0, 133, 251, 169, 4, 133             <131>
730 DATA  252, 162, 0, 169, 1, 129, 251, 24, 165, 251           <236>
740 DATA  105, 1, 133, 251, 165, 252, 105, 0, 133, 252          <022>
750 DATA  201, 8, 208, 235, 96                                  <145>
```

Tastaturpieps

Bei einem Besuch in einer Bank sah ich dort einige größere Computer, die jeden Tastendruck des Bedieners mit einem Piepston quittierten. Dies wollte ich auch beim C 64 nachvollziehen.

Dies sollte unabhängig von einem anderen Programm sein. Daher meine Idee, die Interruptroutine des C 64 zu verändern, da diese 60mal in der Sekunde angesprochen wird, um die Tastatur abzufragen. Dies konnte natürlich nur in Maschinensprache geschehen, da Basic zu langsam wäre. Es ist auch möglich, schon vorhandene Programme damit zu erweitern. Denkbar sind zum Beispiel Textprogramme, Spiele, Programme zum Erlernen des Schreibmaschinenschreibens oder auch nur zur Simulation einer echten Schreibmaschine, die ja auch nicht gerade geräuschlos arbeitet.

Zu Beginn des Programms ($033C-$0348) wird der Interruptvektor verbogen. Er zeigt jetzt auf unsere Routine ($0349-$039E). Da der Computer alle 1/60-Sekunde die Tastatur abfragt, und dazu einen Interrupt auslöst, wird unsere Routine ebenfalls so häufig angesprochen. In dieser Routine wird zuerst überprüft ob überhaupt eine Taste betätigt wurde. Dazu wird der Wert der Speicherstelle $CB in den Akkumulator geladen. Enthält diese den Wert $40, so wurde keine Taste gedrückt und es wird zur normalen Interruptroutine des C 64 gesprungen. Der Wert der Speicherstelle $CB wird zwischengespeichert. Man kann damit bei einem 2. Durchlauf dieser Routine feststellen, ob eine Taste nicht kurz zuvor (t < 1/60 sec.) schon einmal betätigt wurde. Dies dient dazu, daß bei Tasten mit Dauerfunktion nicht unaufhörlich der Pieps ertönt. Denn dies ist erstens entnervend und zweitens verzögert dies den Ablauf der Dauerfunktion so, daß man dabei einschlafen kann. Danach werden die verschiedenen Toneinstellungen vorgenommen. Der Verzögerungsteil ($037F-$0387) dient dazu, den Ton höher zu machen. Ansonsten wäre nur ein Knacken zu hören. Sie können also selbst damit experimentieren und einen für Sie angenehmen Ton einstellen. Das Programm läßt sich durch Drücken der Run/Stop- und der Restore-Taste unterbrechen und mit SYS 49152 reaktivieren.

(Wolfgang Roth / rg)

```
1 DATA 120,162,13,160,192,142,20,3,140,21,3,88,
   96,165,203,201,64,208,8,169                  <097>
2 DATA 203,141,0,193,76,49,234,165,203,205,0,
   193,208,3,76,49,234,169,15                   <227>
3 DATA 141,24,212,169,3,141,5,212,169,242,141,6,
   212,169,26,141,1,212,169                     <255>
4 DATA 5,141,0,212,169,33,141,4,212,160,69,162,
   255,202,208,253,136,208,248                  <105>
5 DATA 169,0,141,4,212,141,5,212,141,6,212,165,
   203,141,0,193,76,49,234,0                    <243>
6 DATA 0,0                                      <021>
10 FOR A=1 TO 99                                <197>
20 READ B : S=S+B :POKE 49151+A,B               <137>
30 NEXT A                                       <225>
40 IF S <> 12111 THEN PRINT CHR$(147)
   : PRINT"FEHLER[SPACE]IN[SPACE]DATAS[SPACE]!"
   : END                                        <000>
50 PRINT CHR$(147) : PRINT"OK[SPACE]!"
   : SYS 49152                                  <046>
60 PRINT"[DOWN]DIES[SPACE]IST[SPACE]DIE[SPACE]
   VERSION,WELCHE[SPACE]AB[SPACE]$C000"
   :PRINT"[DOWN]GESPEICHERT[SPACE]WIRD[SPACE]!";
                       <130>
70 PRINT"[SPACE]UNTERBROCHEN[SPACE]WIRD"
   :PRINT"[DOWN]SIE[SPACE]MIT[SPACE]RUN/STOP
   [SPACE]&[SPACE]RESTORE[SPACE]TASTENDRUCK
   [SPACE]!"                                    <248>
80 PRINT"[DOWN]RESTART[SPACE]MIT[SPACE]SYS49152
   [SPACE]!"                                    <114>
90 END                                          <218>
```

POKE mal wieder

Tastatur statt Joystick

Beim Commodore 64 lassen sich alle Joystick-Funktionen auch über die Tastatur steuern. Hier eine Liste der entsprechenden Tasten:

Joystick Port 1		Joystick Port 2	
Feuer	= SPACE	Feuer	= CTRL + »J«
Links	= CTRL	Links	= CTRL + »D«
Rechts	= »2«	Rechts	= CTRL + »G«
Oben	= »1«	Oben	= CTRL + CRSR RIGHT
Unten	= »§-«	Unten	= CTRL + »A«

(Gunther Knöpfle)

Zeitlupe für den VC 20

Läßt man ein Programm auf dem Bildschirm auflisten, so läuft es in Sekundenschnelle durch. Mit dem Befehl POKE 37877,0 werden nun alle Funktionen des VC 20 extrem verlangsamt, und man kann sich so ein Programm in Ruhe ansehen. Durch Drücken einer beliebigen Taste wird das Listen noch weiter verlangsamt und mit der RUN/STOP-Taste so lange angehalten, wie man die Taste gedrückt hält.

Man kann den Befehl auch innerhalb eines Programms anwenden um zum Beispiel während der Testphase bestimmte Abschnitte sehr langsam und somit nachvollziehbar ablaufen zu lassen.

Mit POKE 37877,72 oder einfach durch gleichzeitiges Drücken von RUN/STOP und RESTORE wird wieder der Normalzustand hergestellt. (Johannes Conrad)

Kommas mit INPUT lesen

Häufig steht man vor dem Problem, einen Textstring mit INPUT zu lesen, der auch Kommas enthalten soll. Versucht man es mit einem normalen INPUT-Befehl, dann meldet der Computer nur EXTRA IGNORED. Mit dem folgenden kleinen Trick kann sowohl der C 64 als auch der VC 20 Texte mit Komma einlesen.

Unmittelbar vor dem INPUT setzt man den Tastenzähler auf 1 und POKEt ein Anführungszeichen in den Tastaturpuffer. Es ergibt sich folgende Programmzeile:

POKE 198,1 : POKE 631,34 : INPUT A$

Probieren Sie's aus. (Udo Stenger)

Verstimmter C 64?

Benutzt man für ein Musikstück die im Commodore 64-Handbuch angegebenen High- und Low-Bytes, um die Töne zu POKEn, dann klingen sie häufig unrein oder »verstimmt«. Das vermeidet man, indem man die Low-Bytes neu festlegt. Man kann sie nach der folgenden Formel berechnen: Low-Byte = Frequenz * 17 - High-Byte * 256

Ist das Ergebnis negativ, dann nimmt man ersatzweise diese Formel: Low-Byte = Frequenz * 17 - (High-Byte - 1) * 256

(Roger Limberg)

VC 20-Tips

Umschalten des VC 20 auf die Grundversion bei eingesteckter Speichererweiterung:

POKE 642,16 : POKE 644,30 : POKE 648,30 : SYS 64824

Mit POKE 55,30 : SAVE »(Name)« kann ein SAVE-Schutz umgangen werden, mit dem viele Programme geschützt sind.

Der Befehl POKE 36867,48 erzeugt eine zusätzliche Zeile unterhalb des normalen Bildschirms, die während des gesamten Programms stehen bleibt und nur über POKE-Befehle zugänglich ist. (Frank Pachollek)

Basic-Programme retten

Ein versehentlich mit »NEW« oder durch einen RESET gelöschtes Programm kann beim VC 20 durch Eingabe der folgenden Befehle im Direktmodus wieder zurückgeholt werden:

POKE 46, PEEK(56) - 1 : POKE 45, PEEK(55) + 247 : CLR »Return«

POKE PEEK(44) * 256 + PEEK(43) + 1, PEEK(44) »Return«

63999 »Return«

FOR I = PEEK(44) * 256 + PEEK(43) TO PEEK(46) * 256 + PEEK(45) : IF PEEK(I) OR PEEK(I + 1) OR PEEK(I + 2) THEN NEXT »Return«

POKE 45, (I + 3) AND 255 : POKE 46, (I + 3) / 256 : CLR »Return«

Unter Umständen erhält man jetzt eine Fehlermeldung, aber das Programm ist jedenfalls wieder da! (Ralf Berle)

C 64 - Bildschirm scrollen

Nach der Eingabe von SYS 59626 wird der gesamte Bildschirm um eine Zeile nach oben verschoben. Umgekehrt geht's aber auch, nämlich mit SYS 59749. Damit werden alle Zeilen ab der aktuellen Cursorposition um eine Position nach unten geschoben. Um den ganzen Bildschirm abwärts zu scrollen, müßte man daher den Cursor in die Bildschirmzeile -1 bringen. Und auch das kann man dem Computer tatsächlich vorgaukeln, und zwar durch POKE 214, 255 : SYS 59749 : SYS 58640. (Michael Wins)

In C 64-Spiele gePOKEt

Hier sind einige interessante POKE-Befehle, mit denen man jeden Highscore überbieten kann. Doch Vorsicht, diese Befehle funktionieren nicht bei allen Versionen dieser Spiele.

* Fort Apocalypse: »POKE 14697,0 : POKE 14760,0 : POKE 36366,0«. Danach hat man beliebig viele Hubschrauber, einen unendlichen Treibstoffvorrat, und der Bonus wird nie erniedrigt.

* Hunchback: »POKE 9521,234 : POKE 9522,234 : POKE 9523,234«. Hier hat man unendlich viele Helden zur Verfügung.

* Neptune: »POKE 7870,60«. Mit diesem POKE hat man auf einen Schlag 60 Taucher.

Jungle Hunt: »POKE 2242,234 : POKE 2243,234«. Der Held hat unendlich viele Leben. (Frank Bastian)

Basic-Programme retten

Die Betriebssystemroutine »Angleich von Koppeladressen» ab Adresse 42291 ermöglicht ein schnelles und einfaches »UNNEW« nach einem versehentlichen »NEW« oder Reset:

POKE 2049,1 : POKE 2050,1 : SYS 42291

Danach kann zumindest wieder geLISTet werden. Ein vollständiges »UNNEW» verlangt allerdings die Korrektur der Zeiger auf den Beginn der Variablen und Felder. Dazu wäre allerdings die Kenntnis der Programmlänge notwendig. Man kann sich aber behelfen, indem man das Programm notfalls in Teilen auf dem Bildschirm aufLISTet und die einzelnen Zeilen mit der RETURN-Taste neu übernimmt. (Gerhard Wagner)

Spezialeffekt

Wenn man beim C 64 in die Speicherstelle 53270 Werte zwischen 0 und 15 schreibt (POKE 53270,x), kann man den Bildschirm um bis zu sieben Bildpunkte nach links oder rechts scrollen lassen. Ist x kleiner als 8, dann scrollt der Bildschirmausschnitt um x Bildpunkte nach links, sonst um x-8 Bildpunkte nach rechts.

POKE 53270,8 stellt den Normalzustand wieder her.

Dieser Trick läßt sich gut bei Action-Spielen als optische Untermalung beispielsweise einer Explosion einsetzen.

(Michael Keukert)

Einige POKEs für den VC 20

Im folgenden ist X immer eine Zahl zwischen 0 und 255. POKE 36865,X: Zentriert den Bildschirm in vertikaler Richtung. Man kann dadurch den Bildschirm nach oben oder unten verschieben. Der Normalzustand wird mit X=38 erreicht. POKE 36864,X: Dieser Befehl ist für die horizontale Bildzentrierung zuständig. Er verschiebt den Bildschirm nach links oder rechts. Der Normalwert ist X=12.

POKE 37879,X: Mit diesem Befehl wird die interne Uhr des VC 20 beeinflußt. Man kann sie schneller oder langsamer laufen lassen. Die letzte Möglichkeit ist besonders beim LISTen interessant. Drückt man nämlich bei verlangsamtem Zeitgeber zusätzlich noch die CTRL-Taste, dann kann man sich einzelne Zeilen fast beliebig lange betrachten. POKE 37879,72 stellt den Normalzustand wieder her.

(Detlef Krischak)

Miner 2049er

Ärgert es Sie auch, daß Sie beim Miner 2049er nie die letzten Bilder vor Augen bekommen? Eigentlich ist das unnötig, denn es gibt einen einfachen Weg, um in die letzten Spielstufen zu gelangen:

Sie brauchen nur entweder die Leertaste bei der Tastatur oder den Feuerknopf am Joystick einige Zeit gedrückt halten. Die augenblickliche Spielstufe wird dann übersprungen und Sie gelangen ins nächste Bild. (Armin Robl)

Directory ohne Programmverlust

Häufig möchte man sich das Directory einer Diskette ansehen ohne das gerade im Speicher befindliche Programm zu zerstören. Wenn man das DOS 5.1 nicht geladen hat, behilft man sich meist mit der zeitaufwendigen Zwischenspeicherung des Programms auf der Diskette. Es geht jedoch auch einfacher und schneller. Geben Sie einfach den folgenden Befehl ein: POKE 44, PEEK(46) + 1

Damit wird der Basic-Anfang auf einen freien Speicherbereich gestellt. Sie können jetzt wie gewohnt mit »LOAD "$",8« das Directory laden und anschließend auflisten.

Mit POKE 44,8 sind Sie dann wieder im eigentlichen Programm. (Heinzpeter Oelkers)

Commodore Joystick verbessert

Der Joystick VIC-1311 für den VC 20 benötigt eine relativ große Hebelbewegung, um die Kontakte zu schließen. Bei Spielen, die eine hohe Reaktionsgeschwindigkeit erfordern, ist diese Eigenschaft sehr ungünstig. Man kann jedoch recht einfach Abhilfe schaffen:

Man entfernt die vier Schrauben an der Unterseite des Gehäuses und hebt den oberen Teil des Joysticks mit der Platine vorsichtig ab. Nun wird die Platine an den Durchtrittsöffnungen der Schrauben mit je einer etwa 1 Millimeter dicken Unterlegscheibe verstärkt. Die vier Unterlegscheiben können mit einem Tropfen Alleskleber (Vorsicht, nicht die Kontakte verkleben !) gegen Verrutschen gesichert werden. Danach wird der Joystick wieder zusammengeschraubt. Wenn Sie alles richtig gemacht haben und insbesondere keine Teile übriggeblieben sind, dann werden die Kontakte des Joysticks nun bei erheblichen kleineren Hebelbewegungen geschlossen. (M. Kunze)

Tips zum C 64

Mit »POKE 808,225« wird die STOP-Taste ausgeschaltet und das Programm kann nicht mehr angehalten werden. »POKE 808,237« schaltet STOP wieder ein.

Eine Repeat-Funktion für alle Tasten wird mit »POKE 650,128« aktiviert. »POKE 650,0« schaltet wieder auf den Normalzustand zurück (Repeat-Funktion nur für die Leertaste und die Cursortasten sowie bei DEL und INST). Mit »POKE 650, 64« wird auch die Wiederholfunktion für diese Tasten abgeschaltet.

»POKE 657, 128« verhindert die Umschaltung zwischen Text- und Grafikmodus mit der Kombination von Shift- und Commodore-Taste. »POKE 657,0« hebt dies wieder auf.

Die Cursorfarbe kann mit »POKE 646,x« festgelegt werden, wobei für »x« der entsprechende Farbcode einzusetzen ist.

»PRINT PEEK(186) ergibt die Gerätenummer des zuletzt angesprochenen Peripheriegerätes. (Thomas Lopatic)

Zwei Tips für den C 64

Die Speicherstellen 57 und 58 enthalten die Zeilennummer der aktuellen Basic-Zeile. Die Abfrage geschieht mit PRINT PEEK(57) + 256 * PEEK(58).

Mit »PRINT PEEK(1)« kann abgefragt werden, ob eine Taste an der Datasette gedrückt ist. Es gibt drei mögliche Werte:

7: Taste gedrückt,
55: keine Taste gedrückt,
39: Taste gedrückt, aber Programmlauf unterbrochen.

Diese Abfragen sind für die benutzerfreundliche Programmierung von Kassettenoperationen recht nützlich.

(Wolfgang Meyer)

VC 20 — Grundversion simuliert

Mit der folgenden kurzen Routine lassen sich die meisten Grundversions- oder +3 KByte-Programme auch mit einer 8 KByte-Erweiterung laden und ausführen:

```
POKE 648, 30 : SYS 64821
POKE 4096, 0 : POKE 44, 16 : NEW
```

Danach wird das Programm ganz normal geladen und gestartet. Da der Bildschirmspeicher durch die kleine Routine an der gleichen Stelle wie in der Grundversion liegt, sind die meisten Grundversions-Programme ohne weitere Änderungen direkt lauffähig. (Sven Jensen)

Maschinensprache »getürkt«

Es gibt eine einfache Methode, um beim VC 20 mit mindestens 8 KByte-Erweiterung aus einem ganz normalen Basic-Programm scheinbar ein Maschinensprache-Programm zu machen.

Zunächst gibt man die folgende Zeile ein:

```
10 SYS 4621
```

Danach folgt im Direktmodus:

```
POKE 43, 1 : POKE 44, 19 : POKE 4864, 0 : NEW
```

Keine Angst, die Programmzeile 10 ist nicht verloren. Jetzt muß das folgende kleine Programm eingegeben werden.

```
10 FOR A = 4621 TO 4643
20 READ D : S = S + D : POKE A, D
30 NEXT
40 IF S§(2039 THEN PRINT "FEHLER IN DATAS"
50 DATA 169, 36, 133, 43, 169, 18, 133, 44
60 DATA 169, 0, 141, 34, 3, 141, 35, 3, 32
70 DATA 89, 198, 76, 174, 199, 0
```

Dieses Programm wird jetzt mit »RUN« gestartet. Anschließend wird im Direktmodus der Basic-Start verschoben:

```
POKE 43, 36 : POKE 44, 18 : POKE 4643, 0 : NEW
```

Damit sind wir schon fast am Ziel. Jetzt kann ein beliebiges Basic-Programm geladen oder geschrieben werden. Die letzte ausführbare Zeile dieses Programms muß immer lauten:

»(Zeilennummer)« POKE 43, 1 : POKE 44, 18 : END

Um das Programm abspeichern zu können, muß man jetzt noch POKE 43, 1 : POKE 44, 18 eingeben. Gibt man nun LIST ein, so sieht man nur 10 SYS 4621. Das auf solch wundersame Art zum »Maschinenprogramm« gewordene Basic-Programm kann jetzt ganz normal gepeichert, geladen und gestartet werden. Allerdings ist jetzt ein hervorragender Listschutz realisiert, zumal im Maschinenspracheteil des Programms noch die RUN/STOP-Taste gesperrt wird. Ein Versuch, das Programm damit abzubrechen, führt stets zu einem völligen »Absturz« des Computers. (A. Kaminski)

»PRINT AT« einfach gemacht

Sehr oft kommt es vor, daß man eine PRINT-Ausgabe an eine bestimmte Bildschirmstelle schreiben will. Üblicherweise arbeitet man in solch einem Falle mit Cursor-Steuerzeichen, was aber einerseits recht langsam ist und andererseits auch die Übersichtlichkeit eines Programms nicht gerade erhöht. Eine Cursorsteuerung über SYS-Kommando unter Ausnutzung der entsprechenden Betriebssystem-Routine ist da schon komfortabler. Aber es geht auch viel einfacher.

Will man beispielsweise einen Text in die 14. Zeile, 5. Spalte schreiben, so geht das auch folgendermaßen:
POKE 214, 13 : PRINT : PRINT TAB(5);"Text"
Adresse 214 enthält nämlich die Zeilenposition des Cursors. Um diese allerdings richtig einsetzen zu können, muß noch ein leerer PRINT-Befehl nachgeschickt werden. Der in Adresse 214 zu POKEnde Zahlenwert ist daher nicht die gewünschte Cursorzeile, sondern der um eins verminderte Wert. (Volker Bühn)

Sequentielle Datei als Basic-Programm laden

Es sind eine Reihe von Anwendungen denkbar, bei denen aus einer sequentiellen Datei auf Diskette oder Kassette ein lauffähiges Basic-Programm erstellt werden soll (Datenfernübertragung, Umwandlung von Textfiles in Basic-Programme). Der folgende Einzeiler macht»s möglich:
OPEN 1, $Gerät(, $Sekundäradresse(, "$Name)" : POKE 812, 238 : POKE 781, 1 : SYS 65478

Dieses Miniprogramm öffnet das File Nummer 1 als Eingabefile (anstelle der Tastatur). Außerdem wird der CLALL-Vektor des Betriebssystems auf ein »RTS« gesetzt, so daß beim Einlesen von Programmzeilen keine Files geschlossen werden.

Deshalb werden nach Eingabe der obigen Befehlszeile von der ausgewählten Datei solange Zeilen eingelesen und sofort im Direktmodus ausgeführt, bis die Betriebssystem-Routine CLRCHN aufgerufen wird (zum Beispiel durch einen Syntaxfehler in den gelesenen Zeilen oder durch GET#1, A$).

Um in den normalen Eingabemodus zurückzukehren, muß nur »POKE 812, 239 : CLR« eingegeben werden.

Zum Ausprobieren: Laden Sie ein beliebiges Basic-Programm und geben Sie danach im Direktmodus ein:
OPEN 1,8,3,"LISTING,S,W" : CMD 1 : LIST : PRINT#1 : CLOSE 1

Dadurch wird das momentan im Speicher befindliche Programm als sequentielles File auf Diskette gespeichert. Datasetten-Besitzer müssen den OPEN-Befehl abändern in »OPEN 1,1,2,"LISTING"«.

Löschen Sie jetzt den Arbeitsspeicher mit »NEW«. Wenden Sie nun unseren Einzeiler auf die sequentielle »LISTING«Datei an:

OPEN 1, 8 ,3, "LISTING,S,R" : POKE 812,238 : POKE 781,1 : SYS 65478

Bei der Datasette geben Sie stattdessen »OPEN 1,1,0, "LISTING" : ...« ein (das Zurückspulen des Bandes nicht vergessen).

Nun werden alle Zeilen des sequentiellen Listings eingelesen, und Sie haben zum Schluß wieder das fertige Programm vor sich.

Natürlich können Sie vor der Anwendung des Einzeilers auch bereits ein Programm im Speicher stehen haben. Da der Computer die Zeilen, die er aus dem sequentiellen File liest, genauso wie Direkteingaben behandelt, werden die neu gelesenen Zeilen mit dem bereits vorhandenen Programm gemischt.

Es ist sogar denkbar, sich spezielle Befehlsfiles für Stapelverarbeitung zu erstellen. Der Computer kann dann ohne weiteres Zutun die vordefinierten Aufgaben durchführen. (Florian Matthes)

VC 20 Tips

POKE 792, 34 : POKE 793, 3 - Nach Drücken der RESTORE-Taste führt der Computer einen Kaltstart aus.

SYS 65511 - Dieser Befehl schließt alle Dateien
POKE 198, 0 : WAIT 198, 1 - Der Computer wartet, bis eine Taste gedrückt wird.
SYS 64802 - Es wird ein vollständiger Kaltstart ausgeführt
SYS 64818 - Kaltstart des VC 20, ist schneller als SYS 64802

POKE 818,34 : POKE 819,253 : POKE 37150,2 - Damit wird das Abspeichern von Programmen verhindert (Kopierschutz). Nach der Eingabe von »SAVE« führt der Computer einen Kaltstart durch. (Thomas Hahn)

Cursor abschalten

Mit dem Befehl "POKE 788,210" wird beim VC 20 der Cursor abgeschaltet. Eine Rücksetzung in den Normalzustand ist mit RUN/STOP-RESTORE möglich. (Dietmar Roudaschl)

Tips & Tricks

MERGE für C 64 / VC 20

Hier ist eine einfache MERGE-Routine zum Verbinden zweier Basic-Programme. Sie kann sowohl für die Floppy als auch für die Datasette (auch mit Turbo-Tape) verwendet werden. Einzige Voraussetzung: Das zweite Programm muß höhere Zeilennummern haben als das erste. Und so wird's gemacht:

1. Sie laden das erste Programm. Dann geben Sie im Direktmodus ein:
PRINT PEEK(43), PEEK(44)
Diese beide Zahlen schreiben Sie sich auf.
2. Sie geben ein:
POKE 43, (PEEK(45) + 256 * PEEK(46) - 2) AND 255 (Return)
POKE 44, (PEEK(45) + 256 * PEEK(46) - 2) / 256 (Return)
Laden Sie nun das zweite Programm. Danach geben Sie ein:
POKE 43, (erste Zahl) : POKE 44, (zweite Zahl) (Return)
Nun befinden sich beide Programme hintereinander im Speicher. (Michael Keukert)

Funktionstastenbelegung unter Simons Basic

Simons Basic bietet ja bekanntlich die Möglichkeit, die Funktionstasten mit beliebigen Zeichenketten zu belegen. Um nun die Funktionstasten nicht jedesmal nach dem Einschalten neu belegen zu müssen, wäre es sinnvoll, die Belegung auf Floppy abspeichern zu können.

Die Funktionstastenbelegung ist bei Simons Basic in dem von Basic nicht erreichbaren Speicherbereich $C64D bis $C74B (50765 bis 51019 dezimal) abgelegt. Mit dem folgenden kleinen Programm wird dieser Speicherbereich als Maschinenprogramm abgespeichert:

```
10 INPUT "Filename "; X$
20 OPEN 5, 8, 5, X$ + »,P,W»
30 A = 50765 : E = 51019
40 H = INT(A / 256) : L = A AND 255
50 PRINT#5, CHR$(L); CHR$(H);
60 FOR I = A TO E
70 PRINT#5, CHR$(PEEK(I));
80 NEXT I : CLOSE 5
```

Mit LOAD "Name",8,1 kann die Funktionstastenbelegung nun jederzeit geladen werden, ohne ein eventuell vorhandenes Basic-Programm zu zerstören. (Uwe Schwarz)

Hilfe für »Turbo Tape«

Das Programm »Turbo Tape» ist ja ein Segen für alle diejenigen, die sich keine Floppy leisten können oder wollen. Es gibt

allerdings einige Maschinenprogramme, die nach dem Gebrauch von »Turbo Tape« abstürzen.

Abhilfe: Nach dem Laden das Programm LISTen und den SYS-Befehl zu Anfang notieren. Nun SYS 64738 und danach den notierten SYS-Befehl eingeben — und schon läuft das Programm. (Andreas Klofanda)

Basic-Programme retten

Ein durch NEW oder durch einen Reset gelöschtes Basic-Programm kann durch Eingabe folgender Zeilen im Direktmodus wieder zurückgeholt werden:

POKE 46, PEEK(56) - 1 : POKE 45, PEEK(55) + 247 : CLR (Return)
POKE PEEK(44) * 256 + PEEK(43) + 1, PEEK(44) (Return)
63999 (Return)
FOR I = PEEK(44) * 256 + PEEK(43) TO PEEK(46) * 256 + PEEK(45) : IF PEEK(I) OR PEEK(I + 1) OR PEEK(I + 2) THEN NEXT (Return)
POKE 45, (I + 3) AND 255 : POKE 46, (I + 3) / 256 : CLR (Return)

Diese »Rettungsmaßnahme« funktioniert sowohl beim VC 20 wie auch beim C 64. (Ralf Berle)

Cursor steuern

Das Betriebssystem des C 64 enthält eine Routine, die man benutzen kann, um den Cursor an eine beliebige Stelle zu setzen. Geben Sie doch mal folgendes ein:

POKE 214, (Zeile) : POKE 211, (Spalte) : SYS 58640 : PRINT "TEXT" (Michael Keukert)

Und noch ein Tip

Der FORMULAR TOO COMPLEX - Error ist sehr unangenehm, da sich das Programm danach oft nicht mehr listen läßt. Nach Eingabe von POKE 24,0 verhält sich der Computer aber wieder normal. (Roger Limberg)

Einige WAIT-Befehle

Folgende Befehle warten auf spezielle Tasten:
WAIT 198, 1 wartet auf beliebige Taste
WAIT 653, 1 Wartet auf Shift
WAIT 653, 2 Wartet auf Commodore-Taste
WAIT 653, 4 Wartet auf CNTRL

(Michael Keukert)

Tips & Tricks

Listschutz

Möchte man ein Programm mit einem einfachen Listschutz versehen, so verfährt man folgendermaßen:

1. Man ergänzt die Zeile, ab der der Listschutz wirksam werden soll, mit ":REM".
2. Man fährt mit dem Cursor auf das zweite Anführungszeichen und drückt fünfmal die Taste INST.
3. Nun wird ebenfalls fünfmal die Taste DEL gedrückt, so daß zwischen den Anführungszeichen fünf reverse T stehen. ..list
4. Zuletzt bewegt man den Cursor hinter das zweite Anführungszeichen und drückt die Tastenkombination SHIFT und L. Anschließend RETURN nicht vergessen.

Wenn nun versucht wird, das Programm zu listen, gelangt der Computer nur bis zu der Zeile, in der der Listschutz steht und bricht dann den Vorgang mit »Syntax Error« ab.

(Thomas Lopatic)

C 64 beschleunigt

Für alle diejenigen C 64 - Besitzer, denen die Bewegung des Cursors bisher zu langsam war, gibt es einen speziellen POKE:

Mit POKE 56325,5 wird der Cursor rasend schnell und flitzt bei Betätigung der Cursortasten nur noch so über den Bildschirm. Wer's lieber gemütlicher mag, der sollte es stattdessen einmal mit POKE 56325,255 probieren.

(Oliver Bausch)

Text und Grafik mischen: Textomat-Tip

Bei Ihrem Software-Test des Textomat von Data-Becker (Ausgabe 9/84) wurde als gravierender Nachteil angeführt, daß bei einmal gestartetem Ausdruck keine Unterbrechung mehr möglich ist. Ich arbeite in Zusammenhang mit Textomat mit einem Epson-Drucker RX 80 F/T. Um den begonnenen Ausdruck zu unterbrechen, kann man einfach den ON-LINE-Schalter betätigen und den Drucker anschließend ausschalten. Sekunden später meldet sich der Textomat am Bildschirm mit dem zu druckenden Text zurück.

Basic-Programme verbinden

So manch C 64-Besitzer wird es schon geärgert haben, daß sein Computer keinen MERGE-Befehl besitzt. Mit wenig Aufwand ist es aber dennoch möglich, Basic-Programme aneinanderzuhängen:

1. Im Direktmodus »PRINT PEEK(43); PEEK(44)« eingeben und sich die Ergebnisse merken.
2. Das erste Programm normal laden.
3. Erscheint jetzt nach »PRINT PEEK(45)« eine Null oder eine Eins, dann geben Sie »POKE 43, 256 + PEEK(45) - 2 : POKE 44, PEEK(46) - 1 : NEW« ein. Im anderen Fall wird »POKE 43, PEEK(45) - 2 : POKE 44, PEEK(46) : NEW« eingegeben.
4. Nun wir das anzuhängende Programm geladen (Achtung! Das anzuhängende Programm muß die höheren Zeilennummern haben).
5. Jetzt POKEn Sie in die Speicherstellen 43 und 44 die zu Anfang gemerkten Werte.

Beide Programme sind nun verbunden und können ganz normal gehandhabt werden. Wichtig bei der ganzen Prozedur ist, daß keine Variablen definiert werden, da das MERGEn sonst nicht richtig funktioniert. (Thomas Lopatic)

POKEs für den C 64

Mit POKE 775, 1 ist ein (fast) perfekter Listschutz aktiviert. Auch ein SAVE-Schutz ist mit wenig Aufwand möglich: POKE 801,0 : POKE 802,0 : POKE 818,165. Nach diesen drei POKE-Befehlen kann das Programm weder auf Kassette noch auf Diskette kopiert werden. Schließlich gibt es noch eine Möglichkeit, die RUN/STOP-Taste abzuschalten, und zwar mit POKE 808,225. Wiedereinschalten ist mit POKE 808,237 möglich. (Thomas Lopatic)

GOTO X für VC 20

Viele schätzen es, viele wünschen es sich: Einen berechneten GOTO-Befehl auf einen variablen Ausdruck anstelle einer Zeilennummer. Hier ist eine schnelle und sichere Methode, die nur 17 Bytes Speicherplatz benötigt.

Schreiben Sie als erste Programmzeile

1 REM"*********" (mindestens neun Sternchen). Anschließend geben Sie im Direktmodus ein: "PRINT PEEK(43) + PEEK(44) * 256 + 6" (RETURN).

Die daraufhin angezeigte Adresse notieren Sie sich bitte. Ohne Erweiterung müßten Sie den Wert 4103 erhalten haben, mit 3 KByte Erweiterung 1031 und ab 8 KByte Erweiterung 4615.

Jetzt POKEn Sie ab der notierten Adresse bitte folgende Werte ein: 32, 138, 205, 32, 247, 215, 76, 163, 200.

Wenn Sie nun die erste Zeile (mit dem REM) aufLISTen, sehen Sie einige Grafikzeichen. Diese stellen ein kurzes Maschinenspracheprogramm dar, das einen mathematischen Ausdruck in einen ganzzahligen Wert umrechnet. Diese Zeile

muß immer die erste Programmzeile sein und darf auch nicht mehr geändert werden. Das übrige Programm kann natürlich wie gewohnt editiert werden.

Sie haben jetzt im Programm einen simulierten GOTO X - Befehl zur Verfügung, der mit SYS (Adresse) X aufgerufen wird. Für Adresse müssen Sie die anfangs notierte Adresse einsetzen (Klammern nicht vergessen). Für X kann ein beliebiger arithmetischer Ausdruck stehen wie zum Beispiel 5, A, A+2, C+D/SQR(9) oder PEEK(5).

Der neue Befehl hat im übrigen die gleichen Auswirkungen wie der normale GOTO-Befehl. Ist eine Zeilennummer nicht vorhanden, gibt es daher ebenfalls einen »UNDEF'D STATEMENT ERROR». (Thomas Maul)

POKEs für den 64'er

POKE 775,200 Listschutz ein
POKE 775,167 Listschutz aus
POKE 788,49 Run Stop ein
POKE 788,52 Run Stop aus
POKE 808,237 Run Stop Restore ein
POKE 808,225 Run Stop Restore aus
POKE 650,128 Dauerfunktion für alle Tasten
POKE 650,0 Dauerfunktion nur für Space und Cursortasten
POKE 650,64 Dauerfunktion aus für alle Tasten

INPUT ohne Fragezeichen

Die Ausgabe eines Fragezeichens beim INPUT-Befehl kann durch Öffnen einer Tastaturdatei unterdrückt werden:

```
10 OPEN 1,0 : REM Tastaturdatei eröffnen
20 INPUT#1,A$ : REM Einlesen von Tastatur ohne Fragezeichen
30 REM Nicht vergessen, die Datei mit CLOSE 1 wieder zu schließen
```

Zwei Einzeiler

Zahlenkonvertierungen von Dezimal nach Hexadezimal braucht man recht häufig. Hier sind zwei Einzeiler zu diesem Thema:

— Hex $X nach dezimal X

```
10 x=0:fori=1tolen(x$):x0=asc(mid$(
x$,i,1)):x=16*x+x0-48+(x0(64)*7:
next
```

— Dezimal X nach hex $X

```
10  x$=":fori=1to4:x0=x/16:x=x-
int(x0)*16:x$=chr$(48+x-(x(9)*7)
+x$:x=x0:next
```

(Volker Everts)

Listschutz

Einen verblüffenden Listschutz für einzelne Zeilen erhält man, indem man an die eigentliche Programmzeile einen REM-Befehl anhängt und dahinter in Anführungszeichen eine Reihe reverser »T« gefolgt von einem Doppelpunkt und einem beliebigen Text schreibt.

Geben Sie doch einmal folgendes ein:

```
10 PRINT"BAUM":REM19reverse T:
10 PRINT"BLUME"
```

Wenn Sie dieses kleine Programm starten, schreibt der Computer »Baum«, listen Sie aber das Programm, so sehen Sie nur die Zeile 10 PRINT»BLUME«. (Roger Limberg)

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Redaktion: Albert Absmeier, Volker Everts, Georg Klinge, Christian Rogge.
Fremdautoren: Loucewski, Mossavi, Auer, Heine, Dreuw, Mockenhaupt, Materna, Weißenberger, Winkler, Kluge, Kölbach, Haas, Fette, Kusch, Kunz, Förtsch, Russell, Schulz, Grothaus, Sberhut, Hagedorn, Thauer, Schwabe, Bühn, Zell, Roth

Layout: Leo Eder (Ltg.)

Zeichnungen: René Nestler

Auslandsrepräsentation:
Schweiz: Markt & Technik Vertriebs AG, Alpenstraße 14, CH-6300 Zug, Tel. 042-223155, Telex: 862329
USA: M&T Publishing Inc.; 2464 Embarcadero Way, Palo Alto, CA 94303

Vertriebsleitung: Hans Hörl

Anzeigen-Verkaufsleitung: Hannelore Schmidt

Anzeigenverwaltung und Disposition: Michaela Hörl

Verlagsleiter M&T-Buchverlag: Günther Frank

Druck: R. Oldenbourg, Hürderstraße 4, 8011 Kirchheim
Auch Anschrift für Beihefter und Beilagen

Preis: Das Einzelheft kostet DM 14,—

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Plieninger Straße 100, 7000 Stuttgart 80 (Möhringen), Telefon (0711) 72004-0

Verantwortlich: Für redaktionellen Teil: Albert Absmeier
Für Anzeigen: Hannelore Schmidt

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0, Telex 5-22052

Aktionäre, die mehr als 25% des Kapitals halten: Otmar Weber, Ingenieur, München; Carl-Franz von Quadt, Betriebswirt, München. Aufsichtsrat: Dr. Robert Dissmann (Vorsitzender), Karl-Heinz Fanselow, Eduard Heilmayr

So machen Sie mehr aus Ihrem COMMODORE 64:

Tips & Tricks

Angerhausen · Englisch
Gerits · Thrun

64
Tips & Tricks

BAND 1
Der Bestseller mit über 70.000 verkauften Exemplaren

Eine Fundgrube für den Commodore 64 Anwender

EIN DATA BECKER BUCH

Hornig · Trapp
Weltner

noch mehr 64
Tips & Tricks

BAND 2

Eine Fundgrube für den Commodore 64 Anwender

EIN DATA BECKER BUCH

DER BESTSELLER – BAND 1

64 Tips & Tricks, das mit über 70.000 Exemplaren meistverkaufte DATA BECKER BUCH, ist eine hochinteressante Sammlung von Anregungen zur fortgeschrittenen Programmierung des COMMODORE 64, POKE's und andere nützliche Routinen, interessanten Programmen. Aus dem Inhalt: 3D-Graphik in BASIC – Farbige Balkengraphik – Definition eines eigenen Zeichensatzes – Tastaturbelegung – Simulation der Maus mit einem Joystick – BASIC für Fortgeschrittene – C-64 spricht deutsch – CP/M auf dem COMMODORE 64 – Druckeranschluß über den USER-Port – Datenübertragung von und zu anderen Rechnern – Synthesizer in Stereo – Retten einer nicht ordnungsgemäß geschlossenen Datei – Erzeugen einer BASIC-Zeile in BASIC – Kassettenpuffer als Datenspeicher – Multitasking auf dem COMMODORE 64-POKE's und die Zeropage – GOTO, GOSUB und RESTORE mit berechneten Zeilennummern, INSTR und STRING-Funktion – Repeat-Funktion für alle Tasten. Alle Maschinenprogramme mit BASIC-Ladeprogrammen.

64 Tips & Tricks ist eine echte Fundgrube für jeden COMMODORE 64 Anwender. **64 TIPS & TRICKS, 1984, über 300 Seiten, DM 49.–**

JETZT NOCH MEHR TIPS & TRICKS – BAND 2

Auch der zweite Band von 64 Tips & Tricks dürfte sehr schnell ein Bestseller werden. Das Buch enthält eine Fülle hochkarätiger Programme, Anregungen und Routinen: ein umfangreiches Kapitel über Softwareschutz – Befehlserweiterungen und wie man sie macht – Tips & Tricks zur Programmierung von Superspielen – Zeiger und deren Manipulation – mehr übers Interrupt-Handling mit vielen Beispielen – erweiterte Hardware-Möglichkeiten – Betriebssystem ins RAM kopieren und dort manipulieren – sowie viele weitere Programme, Befehlserweiterungen und nützliche Routinen. Wer gerne programmiert und mehr wissen will über den COMMODORE 64, der braucht dieses neue Buch.

64 TIPS & TRICKS Band 2, ca. 250 Seiten, DM 39,–

Erscheint: Dezember '84

DATA BECKER's GROSSE PROGRAMM-SAMMLUNG ZUM COMMODORE 64, 250 Seiten, DM 49,–

DER COMMODORE 64 UND DER REST DER WELT, 220 Seiten, DM 49,–

DAS TRAININGSBUCH ZU SIMON's BASIC, 380 Seiten, DM 49,–

COMMODORE 64 FÜR TECHNIK UND WISSENSCHAFT, 300 Seiten, DM 49,–

DAS IDEENBUCH ZUM COMMODORE 64, 240 Seiten, DM 29,–

Diese und viele weitere DATA BECKER BÜCHER gibt's im Buchhandel, im Computerfachhandel und in den Warenhäusern. Dort gibt's auch den kostenlosen, großen DATA BECKER Katalog mit der großen Buch- und Softwareauswahl rund ums Thema Computer. Katalog auch kostenlos direkt von DATA BECKER.

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1

Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten

☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name und Adresse bitte deutlich schreiben